



Titre: Étude sur l'apprentissage d'une approximation du quantron
Title:

Auteur: Simon de Montigny
Author:

Date: 2007

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Montigny, S. (2007). Étude sur l'apprentissage d'une approximation du quantron
Citation: [Master's thesis, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/7970/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7970/>
PolyPublie URL:

**Directeurs de
recherche:** Richard Labib
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

ÉTUDE SUR L'APPRENTISSAGE
D'UNE APPROXIMATION DU QUANTRON

SIMON DE MONTIGNY
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
AVRIL 2007

© Simon de Montigny, 2007.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-29226-6
Our file *Notre référence*
ISBN: 978-0-494-29226-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ÉTUDE SUR L'APPRENTISSAGE
D'UNE APPROXIMATION DU QUANTRON

présenté par : DE MONTIGNY Simon

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. GAGNON Michel, Ph.D., président

M. LABIB Richard, Ph.D., membre et directeur de recherche

M. ADJENGUE Luc, Ph.D., membre

À toutes les étoiles qui ont guidé mon chemin :
à mes parents, à mes professeurs,
et à Marie-ève.

REMERCIEMENTS

Mes remerciements sincères vont à mon directeur Richard Labib pour son soutien et son écoute pendant la réalisation de ce projet. Il m'a permis de développer mon autonomie et mes capacités de planification en me laissant beaucoup de liberté dans la manière de mener mes recherches.

Je tiens également à remercier ma famille et mes amis pour leurs encouragements et pour leur présence dans ma vie.

RÉSUMÉ

Cette recherche porte sur l'approximation d'un nouveau type de neurone artificiel, le quantron. Le quantron possède une fonction discriminante complexe, mais cette dernière peut être simplifiée en utilisant des polynômes quadratiques pour approximer les signaux internes du neurone. Nous proposons deux approximations quadratiques qui permettent d'analyser la fonction discriminante du quantron. Nous avons établi deux théorèmes d'apprentissage pour l'apprentissage des approximations du quantron, et nous avons étudié la performance de ces approximations dans la reproduction des surfaces discriminantes originales du quantron. Nous avons démontré de manière empirique la possibilité d'entraîner la meilleure des deux approximations avec un algorithme de descente du gradient. Finalement, nous abordons la problématique de l'apprentissage dans un réseau de quantrons, et nous proposons une modification qui donne une fonction d'activation continûment dérivable pour le quantron.

ABSTRACT

The subject of this research is the approximation of a new type of artificial neuron, the quantron. Since the analysis of its discriminant function is difficult, we propose to approximate the internal signals of the neuron to yield a simplification of the discriminant function. Two quadratic approximations are obtained, along with two convergence theorems for the simplest approximation. After comparing the results of both approximations to the original quantron, we develop a gradient-based learning algorithm for the best approximation. Finally, we illustrate a problem arising in learning with a network of quantrons, and we propose a modification that yields a smooth activation function for the quantron.

TABLE DES MATIÈRES

| | |
|--|------|
| DÉDICACE | iv |
| REMERCIEMENTS | v |
| RÉSUMÉ | vi |
| ABSTRACT | vii |
| TABLE DES MATIÈRES | viii |
| LISTE DES TABLEAUX | x |
| LISTE DES FIGURES | xi |
| LISTE DES SIGLES ET ABRÉVIATIONS | xii |
| LISTE DES ANNEXES | xiv |
| CHAPITRE 1 INTRODUCTION | 1 |
| CHAPITRE 2 REVUE DE LITTÉRATURE. | 6 |
| 2.1 Caractéristiques des modèles de neurones | 8 |
| 2.2 Problème d'apprentissage | 9 |
| CHAPITRE 3 ANALYSE MATHÉMATIQUE DU QUANTRON | 11 |
| 3.1 Discrétisation du quantron | 14 |
| 3.2 Approximation continue de $S(t)$ | 17 |
| 3.2.1 Approximation des PPS | 21 |
| 3.2.2 Approximation des signaux d'entrée | 30 |
| CHAPITRE 4 ALGORITHMES D'APPRENTISSAGE | 42 |
| 4.1 Apprentissage par les fonctions discriminantes | 44 |
| 4.2 Expérience 1 : Fonctions discriminantes | 59 |
| 4.2.1 Expérience 1A : Application du théorème 1 | 60 |

| | | |
|----------------------------------|--|----|
| 4.2.2 | Expérience 1B : Étude comparative des fonctions discriminantes | 64 |
| 4.3 | Apprentissage par la descente du gradient | 70 |
| 4.4 | Expérience 2 : Descente du gradient | 73 |
| 4.4.1 | Détails d'implémentation | 73 |
| 4.4.2 | Résultats | 73 |
| CHAPITRE 5 RÉSEAUX DE QUANTRONS | | 79 |
| 5.1 | Neurones silencieux | 79 |
| 5.2 | Modification à la sortie du quantron | 84 |
| CHAPITRE 6 ANALYSE ET DISCUSSION | | 88 |
| 6.1 | Approximations polynomiales | 88 |
| 6.2 | Apprentissage par les fonctions discriminantes | 89 |
| 6.3 | Apprentissage par la descente du gradient | 89 |
| 6.4 | Réseaux de quantrons | 90 |
| 6.5 | Discussion | 90 |
| CHAPITRE 7 CONCLUSION | | 92 |
| RÉFÉRENCES | | 94 |
| ANNEXES | | 99 |

LISTE DES TABLEAUX

| | | |
|-------------|--|-----|
| Tableau 4.1 | Résultats moyens pour différentes valeurs de c | 76 |
| Tableau C.1 | Résultats détaillés de l'expérience 2, $c = 1$ | 111 |
| Tableau C.2 | Résultats détaillés de l'expérience 2, $c = 2$ | 111 |
| Tableau C.3 | Résultats détaillés de l'expérience 2, $c = 4$ | 112 |
| Tableau C.4 | Résultats détaillés de l'expérience 2, $c = 6$ | 112 |
| Tableau C.5 | Résultats détaillés de l'expérience 2, $c = 8$ | 113 |
| Tableau C.6 | Résultats détaillés de l'expérience 2, $c = 10$ | 113 |
| Tableau C.7 | Résultats détaillés de l'expérience 2, $c = 12$ | 114 |

LISTE DES FIGURES

| | | |
|-------------|--|----|
| Figure 1.1 | Surface discriminante d'un perceptron (gauche) et d'un réseau de perceptrons (droite) | 3 |
| Figure 3.1 | Exemple de PPS du quantron | 12 |
| Figure 3.2 | Exemples de signaux d'entrée du quantron | 13 |
| Figure 3.3 | Exemple de fonction d'état du quantron | 14 |
| Figure 3.4 | Discrétisation du PPS du quantron | 16 |
| Figure 3.5 | Surfaces discriminantes de la discrétisation du quantron | 18 |
| Figure 3.6 | Résolution du problème XOR | 19 |
| Figure 3.7 | Approximation d'un seul PPS | 21 |
| Figure 3.8 | Exemple de signal d'entrée | 28 |
| Figure 3.9 | Approximation du signal d'entrée | 29 |
| Figure 3.10 | Exemple de $R^2(t)$ | 33 |
| Figure 3.11 | Exemple d'un signal d'entrée et de ses approximations | 38 |
| Figure 3.12 | Exemple de signaux d'entrée et de fonction d'état | 40 |
| Figure 3.13 | Approximations des signaux d'entrée et de la fonction d'état | 41 |
| Figure 4.1 | Surface discriminante de l'expérience 1A | 61 |
| Figure 4.2 | Ensemble d'apprentissage de l'expérience 1A | 62 |
| Figure 4.3 | Surface discriminante résultante | 63 |
| Figure 4.4 | Surfaces discriminantes de la première approximation | 65 |
| Figure 4.5 | Surface discriminante du quantron (gauche) et son approximation par la première méthode (droite) | 65 |

| | |
|---|----|
| Figure 4.6 Surfaces discriminantes du quantron (colonne de gauche) et leur approximation par la deuxième méthode (colonne de droite). Erreurs : 2.81% (haut), 10.97% (centre), 12.70% (bas) | 67 |
| Figure 4.7 Surfaces discriminantes du quantron (colonne de gauche) et leur approximation par la deuxième méthode (colonne de droite). Erreurs : 1.95% (haut), 4.96% (centre), 4.81% (bas) | 68 |
| Figure 4.8 Histogramme des erreurs de reproduction des surfaces discriminantes | 69 |
| Figure 4.9 Surface discriminante de l'expérience 2 | 74 |
| Figure 4.10 Ensemble d'apprentissage de l'expérience 2 | 75 |
| Figure 4.11 Courbes de E et ν | 77 |
| Figure 5.1 Réseau de trois quantrons | 80 |

LISTE DES SIGLES ET ABRÉVIATIONS

| | |
|---|----|
| $g(\cdot)$: fonction discriminante | 2 |
| PPS : potentiel post-synaptique | 9 |
| w : paramètre de poids | 11 |
| s : paramètre de largeur | 11 |
| θ : paramètre de délai | 11 |
| $Q(z)$: probabilité qu'une loi normale centrée réduite soit supérieure à z | 11 |
| a : largeur de la fente synaptique | 11 |
| $\varphi(t)$: fonction représentant le potentiel post-synaptique | 11 |
| M : nombre d'entrée d'un quantron | 12 |
| N : nombre de PPS par entrée d'un quantron | 12 |
| $R(t)$: signal d'entrée d'un quantron | 12 |
| $S(t)$: fonction d'état d'un quantron | 12 |
| x : valeur d'entrée d'un quantron | 12 |
| Γ : seuil d'un quantron | 13 |
| $\varphi_n(t)$: approximation numérique de $\varphi(t)$ | 15 |
| $u(\cdot)$: fonction Heaviside | 16 |
| IEC : intégrale de l'erreur au carré | 20 |
| ESC : erreur du sommet au carré | 20 |
| $p(t)$: polynôme d'approximation | 21 |
| $f(t)$: approximation polynomiale de $S(t)$ | 25 |

| | |
|---|----|
| a_i : coefficients de $f(t)$ | 25 |
| $[\cdot]$: fonction partie entière | 32 |
| t_m : temps auquel le maximum de $f(t)$ est atteint | 42 |
| f_m : valeur que prend $f(t)$ en t_m | 43 |
| α : valeur de sortie du quantron | 43 |
| e_{ij} : coefficients de la fonction discriminante du quantron | 46 |
| Φ_j : transformations d'une Φ -machine | 47 |
| $h(\cdot)$: erreur des moindres carrés | 49 |
| E : fonction d'erreur | 70 |
| c : paramètre de la fonction d'erreur | 70 |
| d : valeur désirée | 70 |
| η : taux d'apprentissage | 71 |
| ν : nombre d'exemples mal classés | 74 |
| IEEE-754 : norme de représentation des nombres réels sur ordinateur | 78 |
| $q(\cdot)$: fonction d'activation du quantron | 79 |
| \perp : sortie indéterminée | 81 |
| t_s : approximation de α | 84 |

LISTE DES ANNEXES

| | | |
|----------|--|-----|
| ANNEXE A | LEMME 4 : COEFFICIENTS DÉTAILLÉS | 99 |
| ANNEXE B | EXPÉRIENCE 1A : IMPLÉMENTATION | 108 |
| ANNEXE C | EXPÉRIENCE 2 : RÉSULTATS DÉTAILLÉS | 111 |

CHAPITRE 1

INTRODUCTION

Les réseaux de neurones artificiels sont des outils utilisés en ingénierie pour effectuer différentes tâches qui ne se prêtent pas à une modélisation mathématique simple. Puisqu'ils sont inspirés du fonctionnement du cerveau, on a tenté de les appliquer à des problèmes où l'humain performe habituellement mieux que les ordinateurs, comme la reconnaissance de formes (identifier correctement un code postal écrit à la main : LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., et al., 1990) et la prédiction du mouvement (mouvement des bras : Miyamoto, Kawato, Setoyama et Suzuki, 1988).

Les premiers modèles de neurone artificiel sont une grossière approximation des vrais neurones biologiques, mais ils arrivent tout de même à résoudre des problèmes assez complexes. Le développement de neurones artificiels plus réalistes permet de créer des réseaux avec une plus grande puissance de calcul (Maass, 1997 ; Labib, 1999).

Le sujet de cette recherche, le quantron, est un neurone artificiel dont le fonctionnement reproduit l'interaction des neurotransmetteurs dans le cerveau. Nous aborderons l'étude du quantron dans le cadre de la théorie de la reconnaissance de formes.

La reconnaissance de formes est le domaine qui étudie le problème suivant : com-

ment identifier des objets avec un ordinateur? Une des étapes importantes de ce problème consiste à différencier des objets de différents types. On appelle *classification* l'opération qui consiste à assigner un type à un objet. Si on considère des classes d'objets prédéfinies, on parle de *classification supervisée*. Lorsque aucune classe n'est précisée à l'avance, et qu'on se base sur les objets eux-mêmes pour choisir les classes, on effectue de la *classification non supervisée*.

La classification se base habituellement sur certaines caractéristiques des objets traités. Dans le monde réel, les objets d'une même classe ne sont pas tous identiques. Il faut donc tolérer une certaine variabilité sur les caractéristiques des objets; on parle alors de l'approche statistique au problème de reconnaissance de formes. On utilise donc le concept de *variable aléatoire*, issu de la théorie des probabilités, pour représenter les caractéristiques des objets. Lorsqu'on ne considère pas la variabilité des objets, on parle plutôt d'une approche déterministe. Ces deux approches sont souvent complémentaires, car on peut développer des techniques de classification d'une manière déterministe, et ensuite étudier leur comportement du point de vue probabiliste.

Dans ce travail, nous étudierons le qantron comme outil de classification supervisée, en utilisant une approche déterministe. Nous utiliserons deux classes correspondant aux deux états du qantron : soit le qantron est actif, soit il est passif.

La reconnaissance de formes se base sur le concept de *fonction discriminante*. De manière formelle, on définit les caractéristiques d'un objet comme un vecteur, noté \vec{x} . La fonction $g(\vec{x})$ est une fonction discriminante pour les classes A et B si les deux conditions suivantes sont remplies : $g(\vec{x}) \geq 0$ si et seulement si \vec{x} appartient à A ,

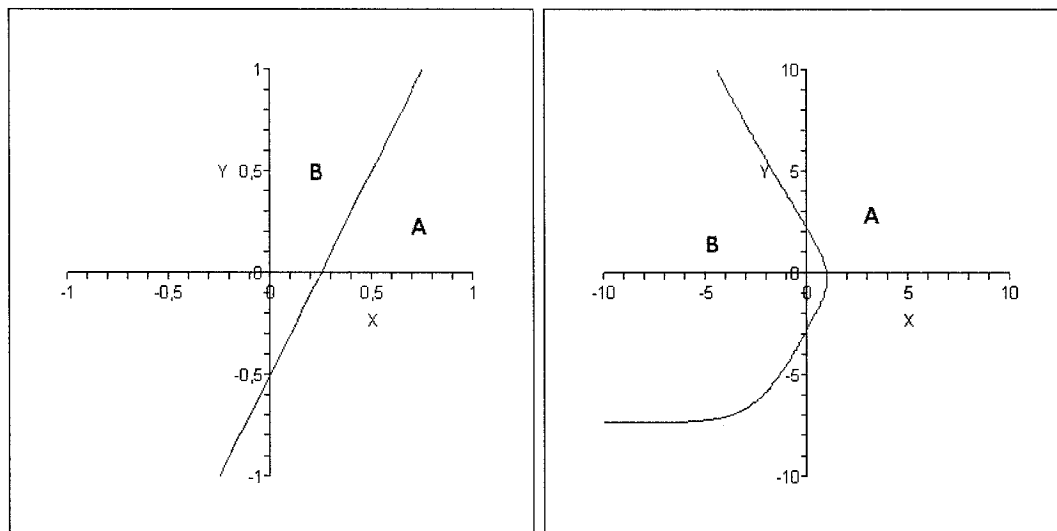


Figure 1.1 – Surface discriminante d'un perceptron (gauche) et d'un réseau de perceptrons (droite)

et $g(\vec{x}) < 0$ si et seulement si \vec{x} appartient à B . On appelle *surface discriminante* l'ensemble des valeurs de \vec{x} satisfaisant l'égalité $g(\vec{x}) = 0$. Lorsque la fonction discriminante d'un problème de classification est inconnue, on cherche à trouver une fonction qui classe correctement le plus d'objets possibles. Ainsi, on peut tolérer qu'une fonction discriminante classe mal quelques objets.

La figure 1.1 présente la surface discriminante d'un perceptron, l'élément de base utilisé dans les réseaux de neurones artificiels, ainsi que la surface discriminante d'un petit réseau de perceptrons.

Dans le domaine de l'intelligence artificielle (et des réseaux de neurones), la recherche d'une fonction discriminante qui résout un problème de classification est un processus qu'on appelle *apprentissage*. L'apprentissage consiste en la modification de la fonction discriminante lorsqu'elle est confrontée à un objet mal classé. Cette mo-

dification est effectuée en variant certains paramètres de la fonction. Puisque cette opération suppose la connaissance à priori de la classe des exemples étudiés, on parle d'*apprentissage supervisé* (en référence à la classification supervisée). Un *apprentissage non supervisé* peut être effectué si on recherche une certaine régularité entre les différents types d'objets (les méthodes statistiques peuvent aider à définir des classes à partir des objets eux-mêmes).

L'apprentissage supervisé dans les réseaux de neurones se base habituellement sur des techniques d'optimisation non linéaire qui traitent des modèles utilisant des fonctions dérivables comme, par exemple, la descente du gradient (Haykin, 1999). Or, la complexité du modèle mathématique du quantron ne permet pas l'utilisation directe de ces méthodes. Ainsi, il est nécessaire de développer des approximations du quantron pour effectuer l'apprentissage de ce dernier. Nous proposons de développer une approximation du quantron en remplaçant, dans son modèle mathématique, certaines fonctions non continûment dérivables par des fonctions qui le sont. Dans le but de développer un modèle qui se traite analytiquement, nous utiliserons des fonctions quadratiques. Ce choix est justifié par la forme des signaux biologiques qui sont représentés dans le quantron, et par la simplicité de ces fonctions, qui permet de pousser l'étude de manière analytique.

La problématique de cette recherche consiste donc à étudier des approximations polynomiales du quantron comme intermédiaire pour effectuer l'apprentissage du quantron. Les capacités d'apprentissage des approximations devront être établies, et la performance des approximations comme outil de reproduction du quantron devra être évaluée. Nous aborderons également, du point de vue théorique, le problème de l'apprentissage avec un réseau de quantrons.

L'objectif de cette recherche est d'établir les éléments de base nécessaires au développement d'un algorithme d'apprentissage rapide et efficace pour les réseaux de quantrons. Pour ce faire, nous définirons deux approximations polynomiales distinctes pour le quantron, et nous étudierons leur apprentissage d'un point de vue théorique et expérimental. L'analyse des fonctions discriminantes et l'application de la descente du gradient seront nos deux principaux outils d'apprentissage. De plus, la performance de ces approximations sera évaluée à l'aide de la comparaison des surfaces discriminantes produites avec celles du modèle original du quantron.

CHAPITRE 2

REVUE DE LITTÉRATURE

Le quantron est un nouveau type de neurone artificiel issu d'un modèle stochastique du phénomène de diffusion des neurotransmetteurs chimiques (Labib, 1999). Son fonctionnement comporte des similitudes avec d'autres modèles de neurones artificiels : les perceptrons et les neurones impulsionnels.

Le perceptron est l'élément de base utilisé dans l'application des réseaux de neurones classiques (Haykin, 1999). Développé à partir d'une compréhension rudimentaire du fonctionnement du cerveau, le perceptron est un modèle assez limité des neurones biologiques. Par contre, l'entraînement des réseaux de perceptrons peut s'effectuer efficacement par un algorithme appelé "rétro-propagation de l'erreur" (Rumelhart, Hinton et Williams, 1986). Des modifications du modèle de base du perceptron permettent d'obtenir des réseaux de neurones artificiels plus puissants ou plus spécialisés (Durbin et Rumelhart, 1989 ; Zhang et Sarhadi, 1993 ; Ridella, Rovetta et Zunino, 1997 ; Labib et Assadi, 2007). Un des objectifs de cette recherche est d'adapter l'algorithme de rétro-propagation au quantron.

Les neurones impulsionnels résultent d'une modélisation mathématique plus poussée des neurones biologiques (Maass, 1997). Plus puissant qu'un réseau de perceptrons, un réseau de neurones impulsionnels est par contre plus difficile à entraîner (Maass et Schmitt, 1999). Les premiers algorithmes d'apprentissages pour les neu-

rones impulsionnels sont inspirés de l'apprentissage hebbien (Gerstner, Kempter, van Hemmen et Wagner, 1996; Ruf et Schmitt, 1997; Gerstner et Kistler, 2002). L'apprentissage hebbien est habituellement non supervisé, mais il est possible de le modifier pour l'utiliser en mode supervisé. L'algorithme *Spikeprop* consiste en une adaptation de l'algorithme de rétro-propagation aux neurones impulsionnels (Bothe, Kok et La Poutre, 2002). La technique utilisée pour pouvoir effectuer une descente du gradient est la linéarisation de la fonction représentant l'état du neurone autour du point de passage du seuil. Bien que *Spikeprop* contienne certaines limitations, des recherches additionnelles ont abouti à des améliorations (Schrauwen et Van Campenhout, 2004; Booij et Nguyen, 2005; Davis, Erdogmus, Rao et Principe, 2003; McKennoch, Liu et Bushnell, 2006). Notre travail avec le quantron diffère de cette approche, entre autre par l'utilisation de fonctions quadratiques comme base d'approximation.

D'autres approches sont possibles pour entraîner des neurones impulsionnels de manière supervisée. Par exemple, les algorithmes évolutifs, les machines à états liquides et l'algèbre linéaire ont été utilisés pour l'apprentissage des neurones impulsionnels (Pavlidis, Tasoulis, Plagianakos, Nikiforidis et Vrahatis, 2005; Maass, Natschläger et Markram, 2002; Burgsteiner, 2005; Carnell et Richardson, 2005). Les deux premières méthodes pourraient être appliquées avec le quantron, mais elles ne cadrent pas dans l'objectif de ce travail, qui est de développer un algorithme d'apprentissage analytique pour le quantron dans un réseau où l'information est propagée vers l'avant. En effet, les machines à états liquides sont des systèmes avec mémoire (il faudrait donc un réseau récurrent), et les algorithmes évolutifs sont une approche heuristique à l'apprentissage. Quant à l'application de l'algèbre linéaire, elle serait par contre difficile à adapter. En effet, elle est utilisée pour définir l'espace vectoriel

des séquences d'impulsions, et le quantron ne fonctionne pas directement avec de telles séquences.

Puisque le quantron partage des caractéristiques communes avec les perceptrons et les neurones impulsionnels, il est important de bien comprendre la nature de ces modèles. Nous allons donc faire un survol de ces neurones artificiels, ainsi que du problème de leur apprentissage.

2.1 Caractéristiques des modèles de neurones

Le perceptron a été exhaustivement étudié, et ses propriétés sont bien connues. Du point de vue de la reconnaissance des formes, on dit du perceptron qu'il est un classificateur linéaire. Le théorème de convergence du perceptron indique comment modifier les paramètres (aussi appelés poids) d'un seul neurone pour apprendre n'importe quel problème de classification linéairement séparable (Haykin, 1999). L'apprentissage d'un réseau de perceptrons est possible en utilisant une fonction d'activation continûment dérivable appropriée, comme la fonction sigmoïde. Le modèle mathématique du réseau est alors une fonction continue, dont les paramètres inconnus sont les poids du réseau. En définissant une fonction d'erreur pour le réseau, on peut alors effectuer l'apprentissage par diverses méthodes d'optimisation non linéaire. L'algorithme de la rétro-propagation de l'erreur, basé sur la descente du gradient de la fonction d'erreur, est la principale méthode utilisée pour entraîner les réseaux de perceptrons (Rumelhart, Hinton et Williams, 1986). Une telle méthode, qui utilise les caractéristiques locales de la fonction d'erreur, ne peut garantir un apprentissage parfait. En effet, la fonction d'erreur du réseau peut posséder des minima locaux. Finalement, des recherches ont été effectuées sur l'apprentissage de réseaux de perceptrons dont

les fonctions d'activations ou les fonctions d'erreur ne sont pas continûment dérivables (Batruni, 1991 ; Redding et Downs, 1991 ; Bartlett et Downs, 1992 ; Goodman et Zeng, 1994).

D'un point de vue biologique, les neurones impulsionnels sont beaucoup plus réalistes que le perceptron (Maass, 1997). On définit l'état interne d'un neurone comme une somme de signaux (appelés potentiels post-synaptiques ou PPS), qui sont activés quand des impulsions arrivent aux entrées du neurone. Lorsque la somme des signaux dépassent un certain seuil, une impulsion est émise à la sortie du neurone. Souvent, des éléments d'inspiration biologique sont inclus dans les réseaux de neurones impulsionnels, comme un temps de réfraction après l'émission d'une impulsion (Gerstner, 1995).

Le quantron peut être considéré comme un modèle hybride entre un perceptron et un neurone impulsionnel. Il possède des entrées similaires à celles du perceptron, mais il fonctionne avec des potentiels post-synaptiques. La forme de la fonction mathématique de ses potentiels est cependant assez différente de celle des neurones impulsionnels (Labib, 1999).

2.2 Problème d'apprentissage

Dans le contexte des réseaux de neurones, l'apprentissage consiste à déterminer les paramètres du réseau à partir d'un ensemble de données, appelé ensemble d'entraînement (Haykin, 1999). En apprentissage supervisé, on associe à chaque élément de l'ensemble d'entraînement une valeur qui correspond à la réponse désirée du réseau. On fait face à un problème de classification si les valeurs désirées correspondent

à différentes classes assignées aux entrées du réseau. Si les valeurs désirées sont numériques, on parle plutôt d'un problème de régression.

Le principal problème théorique au sujet de l'apprentissage supervisé des réseaux de neurones consiste à déterminer si un certain réseau peut représenter exactement un certain ensemble d'entraînement (il s'agit du *problème de consistance*). Autrement dit, on cherche à vérifier l'existence des valeurs des paramètres du réseau telles que l'erreur sur l'ensemble d'entraînement est nulle. L'analyse de ce problème avec des modèles de neurones plus réalistes, comme les neurones impulsionnels, est souvent complexe. Ainsi, on procède à des simplifications, comme l'utilisation de potentiels rectangulaires (Maass, 1997).

Le problème de consistance pour un simple perceptron peut se résoudre en temps polynomial par la programmation linéaire (Megiddo, 1984). Pour un seul neurone impulsionnel avec des potentiels rectangulaires, on peut montrer que le problème de consistance est NP-complet (Maass et Schmitt, 1999). Ce résultat indique que l'apprentissage avec des neurones impulsionnels est plus difficile qu'avec des perceptrons. Le fonctionnement du quantron, similaire à celui des neurones impulsionnels, laisse croire que l'apprentissage de ce dernier serait également difficile. Il faut cependant noter que la complexité du problème de consistance n'indique pas que le développement d'un algorithme d'apprentissage pour le quantron est impossible. On comprend tout de même que les garanties théoriques sur l'apprentissage du quantron ne peuvent pas être aussi fortes que pour le perceptron.

CHAPITRE 3

ANALYSE MATHÉMATIQUE

DU QUANTRON

Les entrées du quantron forment un vecteur de valeurs réelles \vec{x} . Des considérations biologiques préconisent l'utilisation de valeurs d'entrée positives. Chaque entrée est associée à un signal de plusieurs PPS. Chaque PPS a un paramètre de poids w ($w \in \mathbb{R}$) et un paramètre de largeur s ($s \geq 0$). La fonction utilisée pour le PPS du quantron est :

$$\varphi(t) = \begin{cases} 2wQ\left(\frac{\ln a}{\sqrt{t}}\right) & \text{if } 0 \leq t < s \\ 2w \left[Q\left(\frac{\ln a}{\sqrt{s}}\right) - Q\left(\frac{\ln a}{\sqrt{t-s}}\right) \right] & \text{if } s \leq t < 2s \\ 0 & \text{sinon} \end{cases}, \quad (3.1)$$

où a est une constante représentant la largeur de la fente synaptique du neurone, et où $Q(z)$ est la probabilité qu'une variable aléatoire qui suit une loi normale centrée réduite soit plus grande que z . La figure 3.1 montre un exemple de ce PPS, avec $w = 6$, $s = 2.5$ et $a = 3$. L'équation (3.1) est le résultat de l'analyse de la diffusion des neurotransmetteurs dans la fente synaptique par des processus stochastiques (Labib, 1999). Le quantron possède un autre paramètre : un délai θ ($\theta \geq 0$) qui s'applique au signal d'entrée. Ainsi, les paramètres d'apprentissage du quantron sont un vecteur

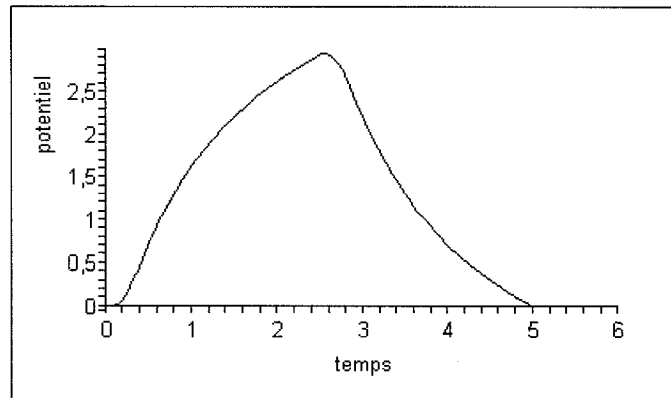


Figure 3.1 – Exemple de PPS du quantron

de poids, un vecteur de largeurs et un vecteur de délais.

Les entrées d'un quantron à M entrées, notées x_1, \dots, x_M , représentent la distance entre le début de deux PPS consécutifs provenant d'une synapse ; il s'agit de l'inverse d'un taux d'arrivée. Pour un quantron qui possède N PPS par entrée, le signal j peut être exprimé par :

$$R_j(t) = \sum_{i=0}^{N-1} \varphi(t - \theta - ix_j) \text{ pour } j = 1, \dots, M. \quad (3.2)$$

La figure 3.2 présente trois exemples du signal d'entrée, pour différentes valeurs de la variable d'entrée.

Finalement, la fonction d'état $S(t)$ du quantron est définie comme suit :

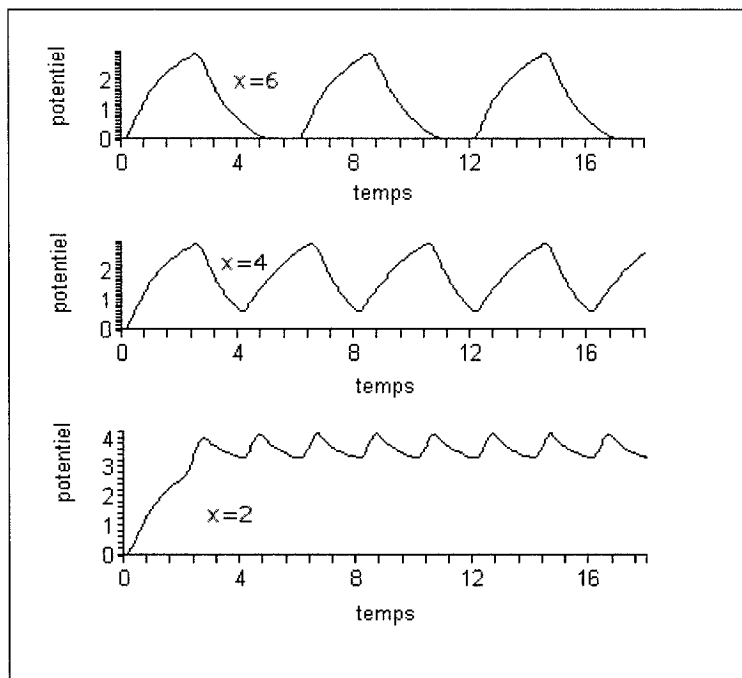


Figure 3.2 – Exemples de signaux d’entrée du quantron

$$S(t) = \sum_{j=1}^M R_j(t). \quad (3.3)$$

La figure 3.3 présente un exemple de la fonction d’état du quantron, avec $w_1 = 4$, $w_2 = -3$, $s_1 = 2.5$, $s_2 = 1.25$, $\theta_1 = 1$, $\theta_2 = 3$, $a = 3$, $N = 6$, $x_1 = 1$ et $x_2 = 0.5$.

Le quantron émet des neurotransmetteurs lorsque sa fonction d’état dépasse pour la première fois un seuil Γ ($\Gamma > 0$). La sortie du quantron est définie comme le temps de premier passage du seuil. Puisqu’il s’agit d’une valeur réelle, le modèle du quantron consiste donc en une fonction à valeurs réelles. La valeur de sortie peut être utilisée comme entrée pour un autre neurone, formant ainsi un réseau. Si le seuil n’est jamais

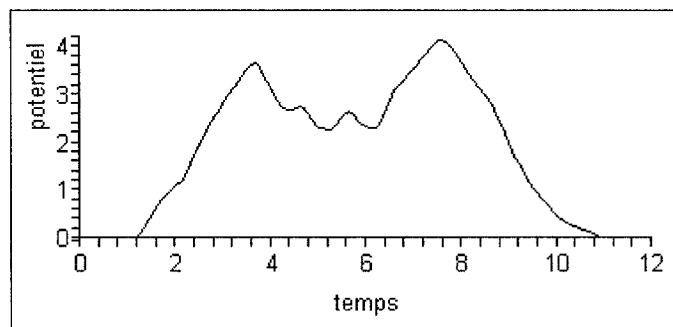


Figure 3.3 – Exemple de fonction d'état du quantron

atteint, il n'y a pas de neurotransmetteurs émis par ce quantron, et alors aucun PPS ne sont générés dans d'autres quantrons par ce dernier. Il faut alors considérer la sortie du quantron comme indéfinie. Nous faisons donc face à une fonction partielle, puisque la sortie du quantron n'est pas définie pour certaines entrées.

On peut aussi utiliser le maximum de la fonction d'état pour décider si le quantron émet ou non. En effet, le quantron émet si et seulement si le maximum de $S(t)$ est supérieur ou égal à Γ . Ce principe peut être utilisé directement avec un seul quantron qui effectue de la classification binaire (les deux classes étant "émet" et "n'émet pas").

3.1 Discrétisation du quantron

Déterminer la valeur de sortie du quantron est un problème complexe. En effet, la fonction d'état du quantron est une somme de fonctions non continûment dérivables dont la forme analytique exacte est inconnue. Par conséquent, le temps de premier passage du seuil (la plus petite solution de l'équation $S(t) = \Gamma$) ne peut être trouvé que par une approximation numérique. Cette approximation consiste à utiliser des PPS plus simples que ceux représentés par l'équation (3.1). De telles simplifications

ont été fréquemment utilisées dans l'étude des neurones impulsionnels (Maass, 1997).

Ainsi, on peut utiliser une fonction rectangulaire :

$$\varphi(t) = \begin{cases} w & \text{si } 0 \leq t < 2s \\ 0 & \text{sinon} \end{cases} \quad (3.4)$$

Cette approximation est grossière, mais elle peut être améliorée. Pour ce faire, nous pouvons considérer une version discrète du PPS du quantron, qui consiste en l'approximation par une fonction en escalier. L'approximation peut se raffiner en augmentant le nombre de points où le PPS est évalué. L'approximation par une fonction en escalier est définie par :

$$\varphi_n(t) = 2w \sum_{k=1}^n \left[Q \left(\frac{\ln a \sqrt{2n}}{\sqrt{2ks} - s} \right) [r_0(t, s) - r_n(t, s)] + Q \left(\frac{\ln a}{\sqrt{s}} \right) r_n(t, s) \right] \quad (3.5)$$

où

$$r_d(t, s) = \begin{cases} 1 & \text{si } \frac{k+d-1}{n} \leq \frac{t}{s} < \frac{k+d}{n} \\ 0 & \text{sinon} \end{cases} \quad (3.6)$$

L'équation (3.5) consiste en une discrétisation de (3.1) avec $2n$ points. La figure 3.4

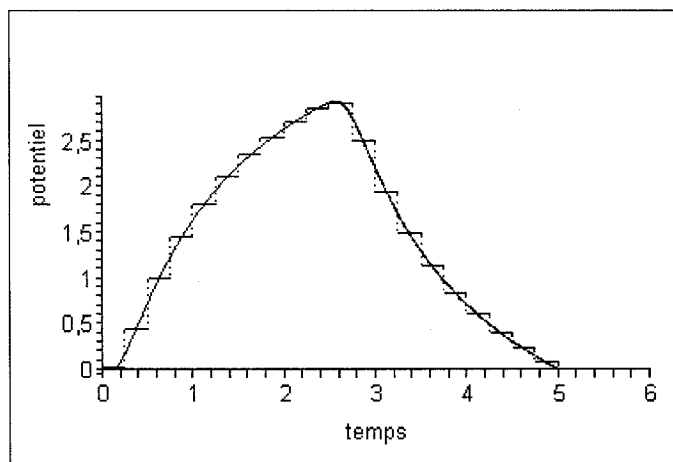


Figure 3.4 – Discrétisation du PPS du quantron avec 20 points d'évaluation

montre cette approximation avec $n = 10$.

Avec la discrétisation du quantron, on peut déterminer plus facilement la valeur de sortie. En effet, il n'y a qu'un nombre fini de points où la fonction d'état peut varier, et on peut facilement déterminer si le seuil est dépassé. En notant la fonction d'état $S(\vec{x}, t)$, pour mettre l'emphase sur le rôle des entrées dans la fonction d'état, nous pouvons exprimer la fonction discriminante du quantron ainsi :

$$g(\vec{x}) = u \left[\sum_{t \in T} u [S(\vec{x}, t) - \Gamma] - 1 \right] \quad (3.7)$$

où $u(\cdot)$ est la fonction Heaviside, et où T est l'ensemble des points où les PPS du quantron changent de valeur. Cette fonction prend la valeur 1 si le quantron émet des neurotransmetteurs, et la valeur 0 si le quantron n'émet rien. La figure 3.5 présente des exemples à deux entrées de la fonction (3.7) avec des PPS de la forme

(3.4). Les axes correspondent aux entrées x_1 et x_2 . La couleur grise correspond aux entrées pour lesquelles $g(x_1, x_2) = 1$, alors que le noir correspond aux entrées pour lesquelles $g(x_1, x_2) = 0$. Nous remarquons que certaines de ces images permettent de résoudre les problèmes linéairement inséparables XOR et XNOR, tel qu'indiqué dans les recherches antérieures sur le quantron (Labib, 1999). La figure 3.6 illustre comment le quantron peut résoudre XOR.

Le problème du XOR est un test classique utilisé pour évaluer la performance des réseaux de neurones. Un réseau de neurones impulsionnels, avec une couche cachée de cinq neurones, et entraîné par Spikeprop permet aussi de résoudre ce problème (Bothe, Kok et La Poutré, 2002). D'ailleurs, un seul neurone impulsionnel peut résoudre XOR, mais le neurone obtenu n'est pas résistant au bruit sur le signal interne (Booij et Nguyen, 2005).

3.2 Approximation continue de $S(t)$

La discrétisation du quantron permet de calculer efficacement la réponse du quantron à un exemple de valeurs d'entrée, mais on ne peut toujours pas utiliser l'algorithme de rétro-propagation. En effet, le maximum de $S(t)$, qui permet de déterminer si un neurone émet ou non, est une fonction non continûment dérivable. On ne peut donc pas utiliser le gradient du maximum de $S(t)$ par rapport aux paramètres d'apprentissage. Ainsi, il faut développer une approximation du maximum de $S(t)$ qui soit continûment dérivable, et qui s'évalue facilement.

Nous proposons deux approximations : le remplacement des PPS par des polynômes quadratiques et le remplacement des signaux d'entrée (une somme de PPS) par

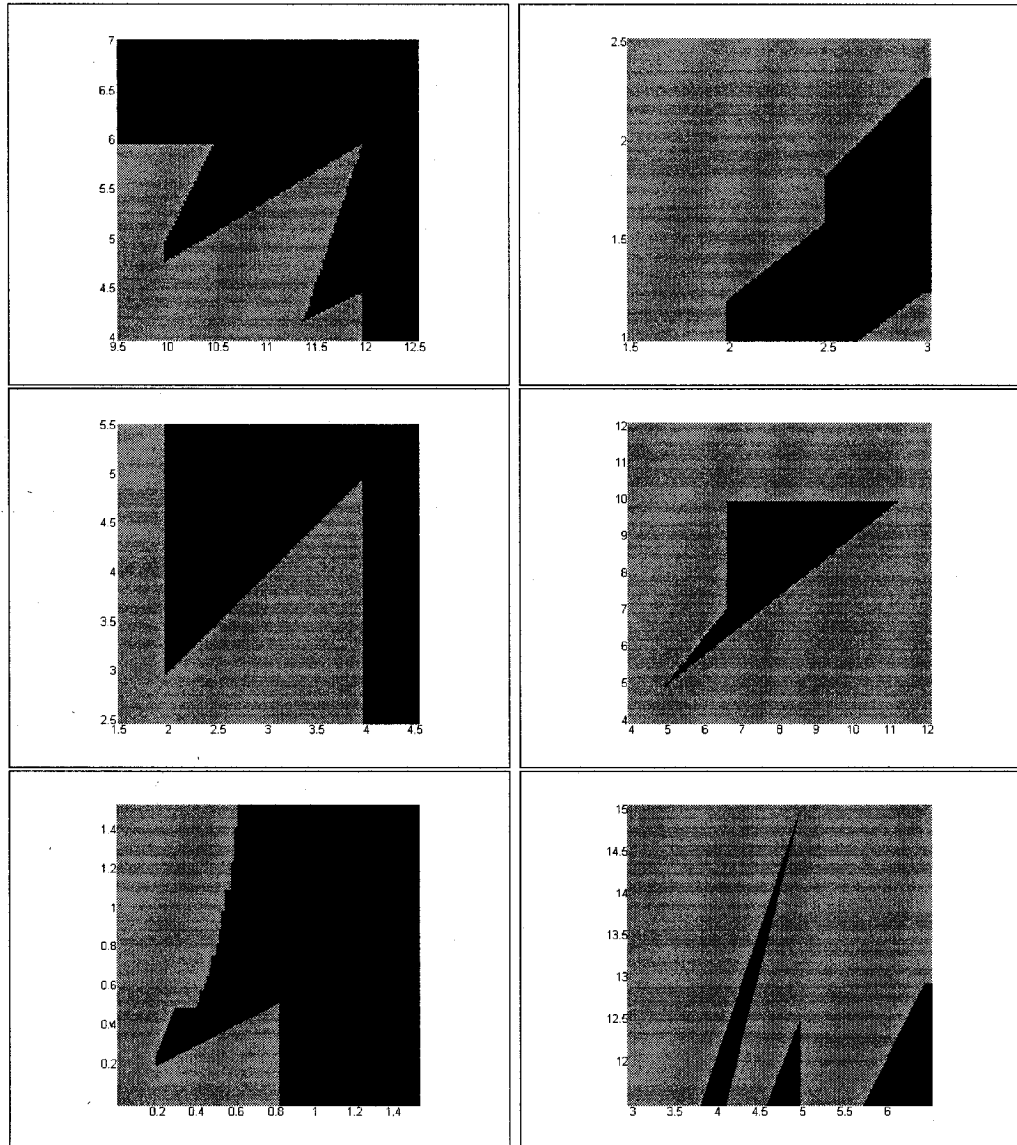


Figure 3.5 – Surfaces discriminantes de la discrétisation du quantron

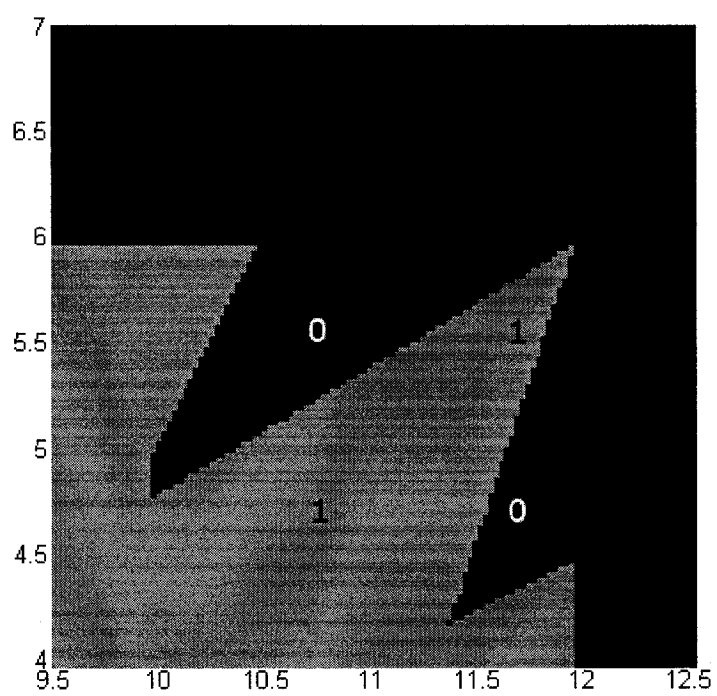


Figure 3.6 – Résolution du problème XOR

des polynômes quadratiques. Pour la première approximation, $S(t)$ est une somme de $M \times N$ polynômes, pour la seconde, $S(t)$ est une somme de M polynômes.

Une bonne approximation polynomiale devrait minimiser une fonction d'erreur représentant la différence entre la fonction étudiée et son approximation. Nous proposons deux fonctions d'erreur distinctes pour chercher de bonnes approximations : l'intégrale de l'erreur au carré (*IEC*), et l'erreur du sommet au carré (*ESC*). L'*IEC* consiste à évaluer l'intégrale, sur un certain intervalle, du carré de la différence entre la fonction étudiée et son approximation. Cette fonction d'erreur permet donc de trouver la représentation la plus approchée de la fonction à approximer. Nous définissons l'*ESC* comme étant le carré de la différence entre le maximum de la fonction étudiée, et le maximum de l'approximation. Dans le contexte de ce travail, une des caractéristiques les plus importants des fonctions utilisées pour représenter différents phénomènes électriques dans le neurone est leur maximum. Ainsi, l'*ESC* guide la recherche d'une approximation en se basant uniquement sur cette caractéristique.

Le calcul des approximations sera basé sur une comparaison avec les PPS rectangulaires, ce qui permettra d'évaluer analytiquement les paramètres des approximations.

Le désavantage de l'utilisation de polynômes est que les signaux d'entrée du quantron peuvent prendre des valeurs négatives même si tous les paramètres w sont positifs, ce qui est impossible dans le modèle original du quantron. Cependant, ces valeurs anormales ne sont pas problématiques, car elles sont habituellement situés à l'extérieur de la région d'intérêt lorsque le maximum de l'approximation de $S(t)$ est près du seuil.

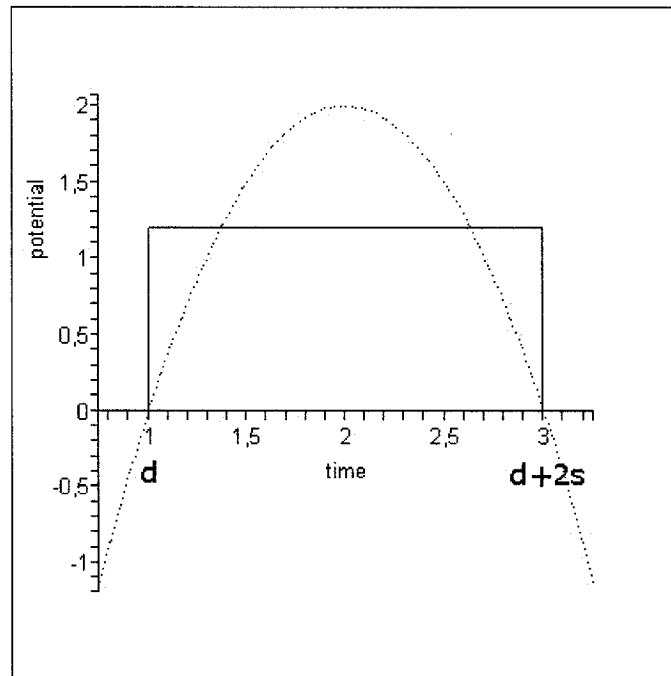


Figure 3.7 – Approximation d'un seul PPS

Dans les deux méthodes d'approximation, les calculs sont effectués pour des signaux excitateurs. La même approche peut être utilisée pour des signaux inhibiteurs en permettant au paramètre w de prendre des valeurs négatives.

3.2.1 Approximation des PPS

Tel qu'illustré dans la figure 3.7, les racines du polynôme seront $t_1 = d$ et $t_2 = d + 2s$, et il s'exprime alors ainsi :

$$p(t) = k [t^2 - (2d + 2s)t + d^2 + 2sd], \quad (3.8)$$

où k est une variable multiplicative dont la valeur sera trouvée en minimisant une fonction d'erreur. L'approximation d'un PPS sera forcément mauvaise hors de l'intervalle $[d, d + 2s]$, sauf si $p(t) = 0$ pour tout t , car le polynôme s'éloigne toujours plus de la valeur du PPS. Cependant, nous remarquons que les valeurs négatives du polynôme sont situées aux temps où le PPS prend une valeur nulle. Ainsi, l'intérêt de cette approximation se situe dans cette correspondance entre le signe des valeurs que prend le polynôme et l'existence d'un potentiel électrique dans le temps.

Nous définissons l'*IEC* ainsi :

$$IEC = \int_d^{d+2s} [\varphi(t) - p(t)]^2 dt, \quad (3.9)$$

soit l'intégrale du carré de la différence entre la fonction d'un PPS rectangulaire du quanton et son approximation. L'intégrale s'effectue sur l'intervalle $[d, d + 2s]$, qui correspond aux valeurs du temps où le PPS rectangulaire prend une valeur non nulle.

Le calcul de cette intégrale donne le résultat suivant :

$$IEC = 2w^2s + \frac{8}{3}ws^3k + \frac{16}{15}s^5k^2. \quad (3.10)$$

La valeur optimale de k est donc donnée par :

$$k^* = -1.25 \frac{w}{s^2}. \quad (3.11)$$

Finalement, le minimum de l' IEC est :

$$IEC^* = \frac{1}{3} w^2 s. \quad (3.12)$$

En contrepartie, nous définissons l' ESC ainsi :

$$ESC = [\max \varphi(t) - \max p(t)]^2, \quad (3.13)$$

ce qui correspond exactement à la définition donnée précédemment. Nous obtenons le résultat suivant :

$$ESC = w^2 + 2ws^2k + s^4k^2. \quad (3.14)$$

La valeur optimale de k est donc donnée par :

$$k^* = -\frac{w}{s^2}. \quad (3.15)$$

Nous trouvons que le minimum de l'*ESC* est nul, ce qui est normal, puisque k^* est telle que $\max \varphi(t) = \max p(t)$.

Puisque les deux valeurs de k^* contiennent le paramètre w , avec pour seule différence une constante multiplicative, nous considérons qu'elles sont équivalentes puisque l'architecture du quantron reste identique si on change l'échelle de valeur des poids. En utilisant $k^* = -\frac{w}{s^2}$, l'approximation polynomiale devient :

$$p(t) = -\frac{w}{s^2} [t^2 - (2d + 2s)t + d^2 + 2sd]. \quad (3.16)$$

Nous déterminons maintenant la forme de la fonction $S(t)$ à partir de l'approximation polynomiale des PPS.

Lemme 1 *La première approximation de $S(t)$ est un polynôme quadratique dont les coefficients peuvent être obtenus analytiquement.*

Preuve En utilisant (3.16) pour approximer le i ème PPS de la j ème entrée x_j , nous avons

$$p_i(t) = -\frac{w_j}{s_j^2} [t^2 - (2\theta_j + 2ix_j + 2s_j)t + (\theta_j + ix_j)^2 + 2s_j(\theta_j + ix_j)]. \quad (3.17)$$

L'approximation polynomiale de $S(t)$, notée $f(t)$, est définie par

$$f(t) = \sum_{j=1}^M \sum_{i=0}^{N-1} p_i(t), \quad (3.18)$$

car il s'agit de la somme de N PPS provenant de M entrées. Nous avons donc

$$f(t) = -a_2 t^2 + a_1 t - a_0, \quad (3.19)$$

où les coefficients a_n ($n = 0, 1, 2$) doivent être déterminés.

Avec la notation suivante :

$$\beta_j = \frac{Nw_j}{s_j^2}, \quad (3.20)$$

le coefficient a_2 s'exprime directement par

$$a_2 = \sum_{j=1}^M \beta_j. \quad (3.21)$$

Pour le coefficient a_1 , nous trouvons

$$\begin{aligned} a_1 &= \sum_{j=1}^M \sum_{i=0}^{N-1} \frac{w_j}{s_j^2} (2\theta_j + 2ix_j + 2s_j) \\ &= \sum_{j=1}^M \beta_j [2\theta_j + (N-1)x_j + 2s_j], \end{aligned} \quad (3.22)$$

et finalement, pour le coefficient a_0 , nous trouvons

$$\begin{aligned} a_0 &= \sum_{j=1}^M \sum_{i=0}^{N-1} \frac{w_j}{s_j^2} [(\theta_j + ix_j)^2 + 2s_j(\theta_j + ix_j)] \\ &= \sum_{j=1}^M \beta_j \left[(N-1) \left(\frac{2N-1}{6} x_j^2 + (\theta_j + s_j)x_j \right) + \theta_j^2 + 2\theta_j s_j \right] \end{aligned} \quad (3.23)$$

Nous avons donc exprimé les coefficients de (3.19) en fonction des entrées et des paramètres du quantron. \square

Les figures 3.8 et 3.9 illustrent respectivement le signal correspondant à une entrée du quantron avec des PPS rectangulaires, et l'approximation de ce signal par des fonctions quadratiques. Les parties négatives des approximations entraînent un maximum qui possède une valeur plus faible pour le maximum de l'approximation. Cependant, nous pouvons ajuster l'échelle des poids au besoin pour régler ce problème, et ce sans changer l'architecture du quantron.

Nous remarquons que le coefficient a_2 est indépendant des variables d'entrée. Si a_2 est négatif, $f(t)$ est convexe est l'approximation n'est pas valide. Il faut donc considérer la condition suivante :

$$\sum_{j=1}^M \beta_j > 0, \quad (3.24)$$

qui sera utile pour définir la fonction discriminante de l'approximation quadratique. Cette condition peut être incluse dans un algorithme d'apprentissage comme contrainte sur les paramètres w_j et s_j .

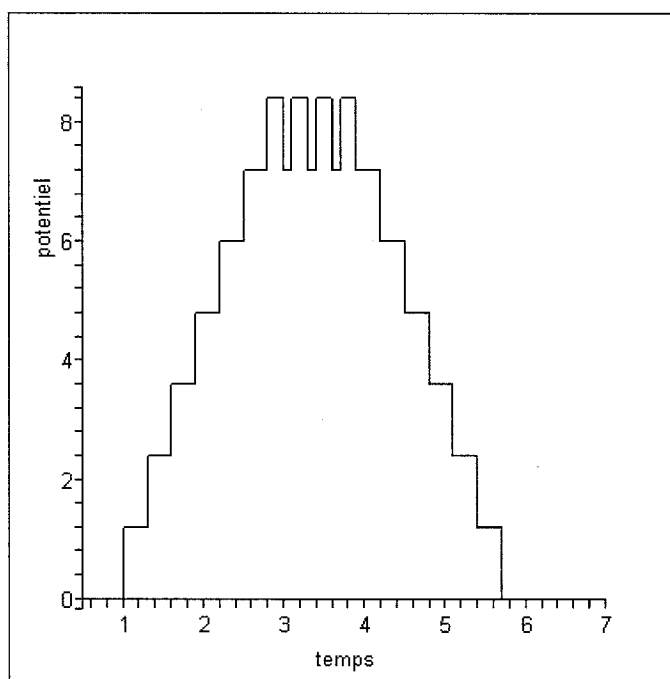


Figure 3.8 – Exemple de signal d'entrée

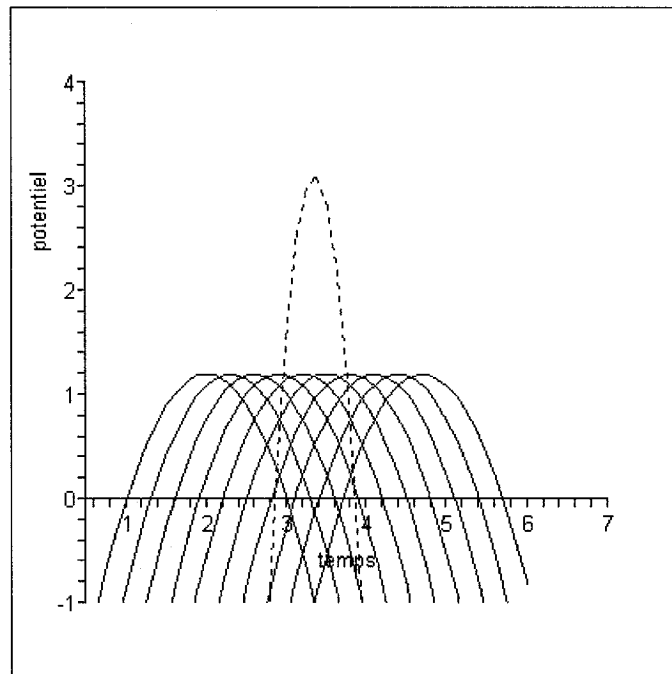


Figure 3.9 – Approximation du signal d'entrée

3.2.2 Approximation des signaux d'entrée

Comme nous l'avons vu précédemment, la somme des PPS d'un signal d'entrée peut être exprimé comme suit :

$$R(t) = \sum_{i=0}^{N-1} \varphi(t - \theta - ix). \quad (3.25)$$

Cette somme prend des valeurs positives (ou nulles) pour $t \in [\theta, \theta + (N - 1)x + 2s]$. De la même manière que dans l'approximation précédente, nous commençons avec le polynôme suivant :

$$p(t) = k [t^2 - (2\theta + (N - 1)x + 2s)t + \theta^2 + (N - 1)\theta x + 2s\theta]. \quad (3.26)$$

L'*IEC* est défini par :

$$\begin{aligned} IEC &= \int_{t_1}^{t_2} [R(t) - p(t)]^2 dt \\ &= I_1 + I_2 + I_3, \end{aligned} \quad (3.27)$$

avec $t_1 = \theta$ et $t_2 = \theta + (N - 1)x + 2s$, et où

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} \int_{t_1}^{t_2} p^2(t) dt \\ -2 \int_{t_1}^{t_2} R(t)p(t) dt \\ \int_{t_1}^{t_2} R^2(t) dt \end{bmatrix}. \quad (3.28)$$

On peut évaluer I_1 directement :

$$I_1 = \frac{k^2}{30} [(N - 1)x + 2s]^5. \quad (3.29)$$

Cependant, la résolution des intégrales I_2 et I_3 est plus délicate. En utilisant (3.25), nous avons

$$I_2 = -2 \int_{t_1}^{t_2} \sum_{i=0}^{N-1} p(t) \varphi(t - \theta - ix) dt. \quad (3.30)$$

En permutant l'intégrale et la somme, et en utilisant (3.4), nous avons

$$I_2 = -2w \sum_{i=0}^{N-1} \int_{\theta+ix}^{\theta+ix+2s} p(t) dt. \quad (3.31)$$

Avec le changement de variable $u = t - \theta$, nous obtenons :

$$I_2 = -4wsk \sum_{i=0}^{N-1} \left[x^2 i^2 + 2sxi - (s + ix) ((N-1)x + 2s) + \frac{4s^2}{3} \right], \quad (3.32)$$

ce qui donne :

$$I_2 = \frac{2}{3}Nwsk [(N-1)(N-2)x^2 + 6(N-1)sx + 4s^2]. \quad (3.33)$$

Finalement, pour résoudre I_3 , nous utilisons la notation suivante :

$$\mu = \min \left\{ N, \left\lfloor \frac{2s}{x} \right\rfloor + 1 \right\}, \quad (3.34)$$

où $\lfloor \cdot \rfloor$ est la fonction partie entière. μ est le nombre maximal de PPS superposés dans $R(t)$. Nous avons donc

$$\max R(t) = \mu w, \quad (3.35)$$

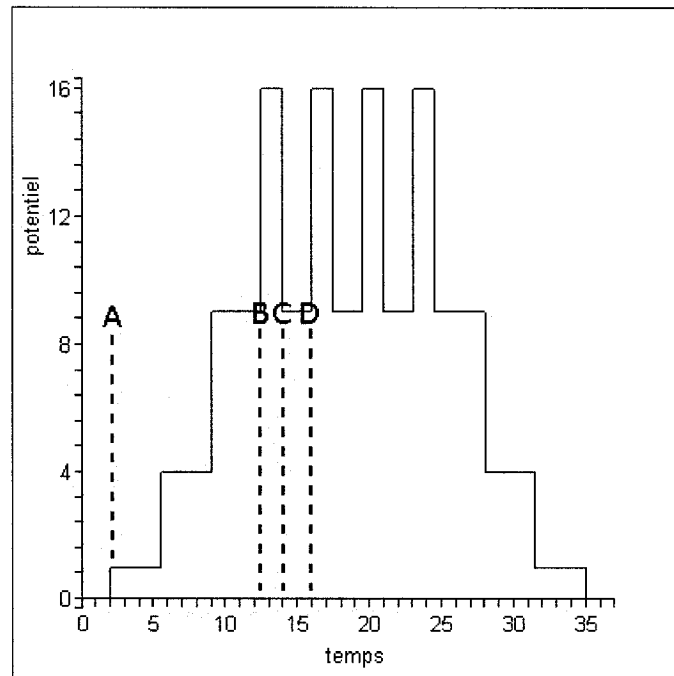


Figure 3.10 – Exemple de $R^2(t)$: un seul signal d'entrée élevé au carré

La figure 3.10 montre un exemple de $R^2(t)$. Les sections constantes des lignes A à B ont toutes une largeur de x . Il y a $\mu - 1$ de ces sections au début et à la fin de $R^2(t)$, avec les hauteurs $w^2, (2w)^2, \dots, [(\mu - 1)w]^2$. Donc, la distance entre les lignes A et B est $(\mu - 1)x$, et puisque la distance entre les lignes A et C est $2s$, la distance entre les lignes B et C est $2s - (\mu - 1)x$. Il y a $N - \mu + 1$ sommets de largeur $2s - (\mu - 1)x$ et de hauteur $(\mu w)^2$. La distance entre les lignes C et D est $\mu x - 2s$. Il y a $N - \mu$ écarts de largeur $\mu x - 2s$ et de hauteur $[(\mu - 1)w]^2$ entre les sommets. Donc, en considérant la hauteur des différentes sections constantes de $R^2(t)$, et en utilisant le changement de variable $u = t - \theta$, nous avons :

$$I_3 = 2x \sum_{i=1}^{\mu-1} (iw)^2 + (N-\mu+1)(\mu w)^2 [2s - (\mu-1)x] + (N-\mu)[(\mu-1)w]^2 (\mu x - 2s). \quad (3.36)$$

Ce résultat est aussi valide pour les valeurs limites $\mu = 1$, $\mu = \frac{2s}{x} + 1$ et $\mu = N$. Nous obtenons donc le résultat suivant :

$$I_3 = w^2 x \mu (\mu - 1) \left[\frac{2\mu - 1}{3} - N \right] + 2s w^2 [\mu + 2\mu N - N] - 2s w^2 \mu^2. \quad (3.37)$$

La valeur optimale de k est donc donnée par :

$$k^* = -10 \frac{N w s [(N-1)(N-2)x^2 + 6(N-1)sx + 4s^2]}{[(N-1)x + 2s]^5}. \quad (3.38)$$

Finalement, le minimum de l'IEC est :

$$IEC^* = I_3 - \frac{10N^2 w^2 s^2 [(N-1)(N-2)x^2 + 6(N-1)sx + 4s^2]^2}{3[(N-1)x + 2s]^5}. \quad (3.39)$$

L'*ESC* est défini par :

$$ESC = [\max R(t) - \max p(t)]^2. \quad (3.40)$$

En utilisant (3.35), nous avons le résultat suivant :

$$ESC = \mu^2 w^2 + 2\mu w \left[\frac{(N-1)x + 2s}{2} \right]^2 k + \left[\frac{(N-1)x + 2s}{2} \right]^4 k^2. \quad (3.41)$$

La valeur optimale de k est donc exprimée par :

$$k^* = -4 \frac{\mu w}{[(N-1)x + 2s]^2}. \quad (3.42)$$

Nous ne pouvons pas utiliser cette valeur de k directement puisque μ n'est pas continue. En supposant que

$$\mu = \left\lfloor \frac{2s}{x} \right\rfloor + 1, \quad (3.43)$$

(en considérant donc les situations où $2s < Nx$) nous utiliserons l'approximation continue suivante :

$$\mu \simeq \frac{2s}{x} + \frac{1}{2}. \quad (3.44)$$

Ce choix est justifié par l'observation suivante : le cas dégénéré (où $2s \geq Nx$) ne permet pas d'effectuer de classification. En effet, si $\mu = N$, le maximum de l'approximation est indépendant de la variable d'entrée du neurone, et alors l'état du quantron est le même pour toutes les valeurs possibles de la variable d'entrée. Pour un quantron à plusieurs entrées, il serait possible d'utiliser $\mu = N$, mais ce choix introduirait des difficultés dans l'entraînement d'un réseau de quantrons. En effet, nous verrons au Chapitre 5 qu'il est possible qu'un neurone, dans un réseau de quantrons, ait des entrées qui ne génèrent aucun PPS pour des valeurs particulières des variables d'entrée du réseau. Ainsi, l'apprentissage de certaines entrées pourrait être problématique dans un réseau de quantrons si $\mu = N$.

En tenant compte de ce choix, nous avons

$$k^* \simeq -2 \frac{w(4s + x)}{x[(N - 1)x + 2s]^2}, \quad (3.45)$$

ce qui donne :

$$ESC^* \simeq w^2 \left[\mu - \left(\frac{2s}{x} + \frac{1}{2} \right) \right]^2, \quad (3.46)$$

et, puisque la différence absolue entre (3.43) et (3.44) est plus petite que $\frac{1}{2}$, nous trouvons que $ESC^* \leq \frac{w^2}{4}$ (si l'inégalité $2s < Nx$ est vérifiée).

Le choix de k n'est pas aussi évident que dans la première approximation. La figure 3.11 illustre un exemple de l'approximation du signal d'entrée avec les deux valeurs de k . Dans cet exemple particulier, ces deux valeurs donnent des approximations similaires, mais cela ne saurait être le cas pour toutes les valeurs possibles des paramètres du quantron. Nous trouvons que le rapport des deux valeurs trouvées est donné par :

$$\frac{5Ns [(N-1)(N-2)x^2 + 6(N-1)sx + 4s^2]}{3x(4s+x)[(N-1)x + 2s]^3}, \quad (3.47)$$

soit le rapport d'un polynôme de degré 2 sur un polynôme de degré 5 (nous considérons ici des polynômes de la variable x). Ainsi, en augmentant la valeur de x , le rapport entre les deux valeurs de k peut devenir très petit, ce qui indique que la différence entre les deux approximations ne saurait être ignorée.

Nous remarquons la borne de ESC^* ne dépend que du paramètre w . Or, nous avons déjà mentionné que l'architecture du quantron ne dépend pas des valeurs

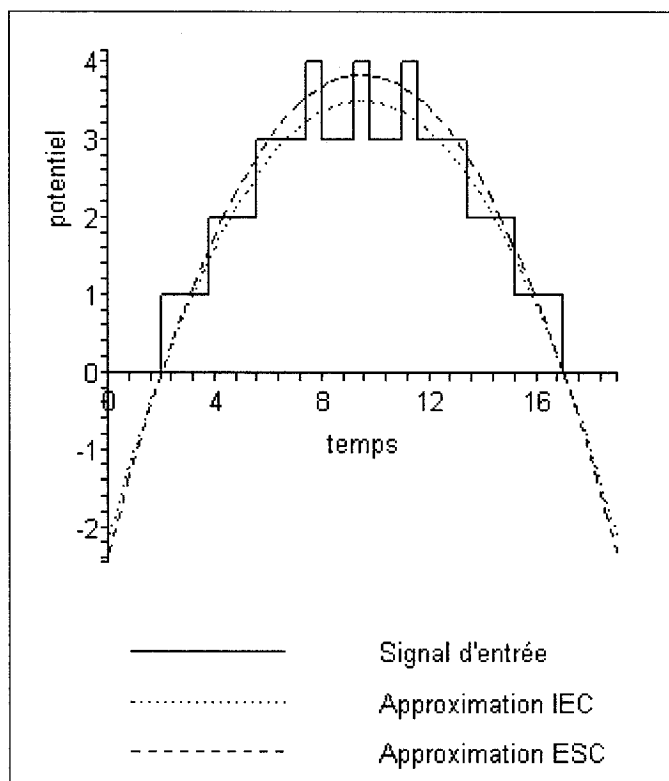


Figure 3.11 – Exemple d'un signal d'entrée et de ses approximations

exactes prises par les poids, et qu'il est possible de changer l'échelle de valeur des poids. Ainsi, du point de vue du fonctionnement du quantron, cette borne semble être une constante. Ceci est intéressant, car nous cherchons à approximer le maximum de la fonction d'état du quantron, et nous avons, au moins, une borne sur l'erreur du maximum des signaux d'entrée. Lorsque les maximums des signaux d'entrée sont rapprochés, nous devrions obtenir une bonne approximation du maximum de la fonction d'état. De plus, d'un point de vue purement algébrique, la valeur de k donnée par l'*ESC* possède une forme plus simple. Nous choisissons donc d'utiliser cette valeur pour la suite de notre analyse.

En substituant (3.45) dans (3.26), l'approximation de $S(t)$ est donnée par

$$f(t) = -a_2 t^2 + a_1 t - a_0, \quad (3.48)$$

où les coefficients a_n ($n = 0, 1, 2$) sont directement donnés par

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^M \frac{2w_j(4s_j+x_j)[\theta_j^2+(N-1)\theta_j x_j+2\theta_j s_j]}{x_j[(N-1)x_j+2s_j]^2} \\ \sum_{j=1}^M \frac{2w_j(4s_j+x_j)[2\theta_j+(N-1)x_j+2s_j]}{x_j[(N-1)x_j+2s_j]^2} \\ \sum_{j=1}^M \frac{2w_j(4s_j+x_j)}{x_j[(N-1)x_j+2s_j]^2} \end{bmatrix}. \quad (3.49)$$

La figure 3.12 présente un exemple de signaux d'entrée et de fonction d'état du quantron, et la figure 3.13 montre les approximations de ces signaux.

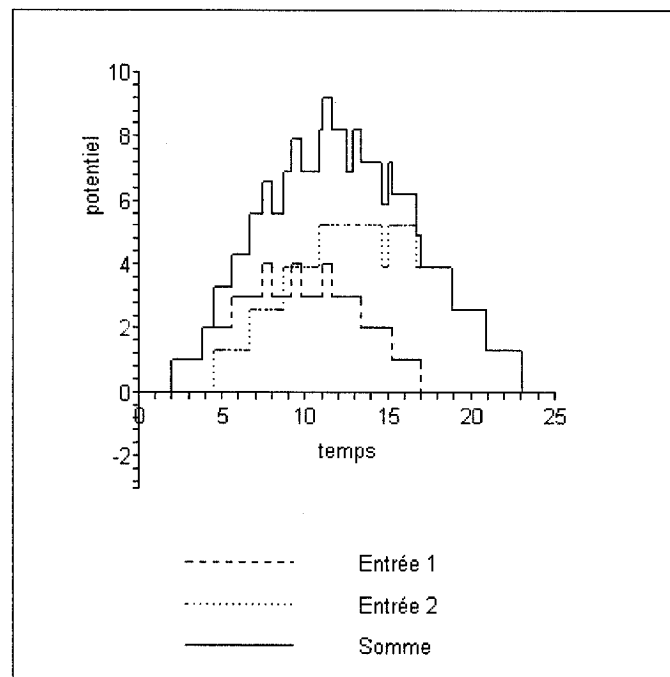


Figure 3.12 – Exemple de signaux d'entrée et de fonction d'état

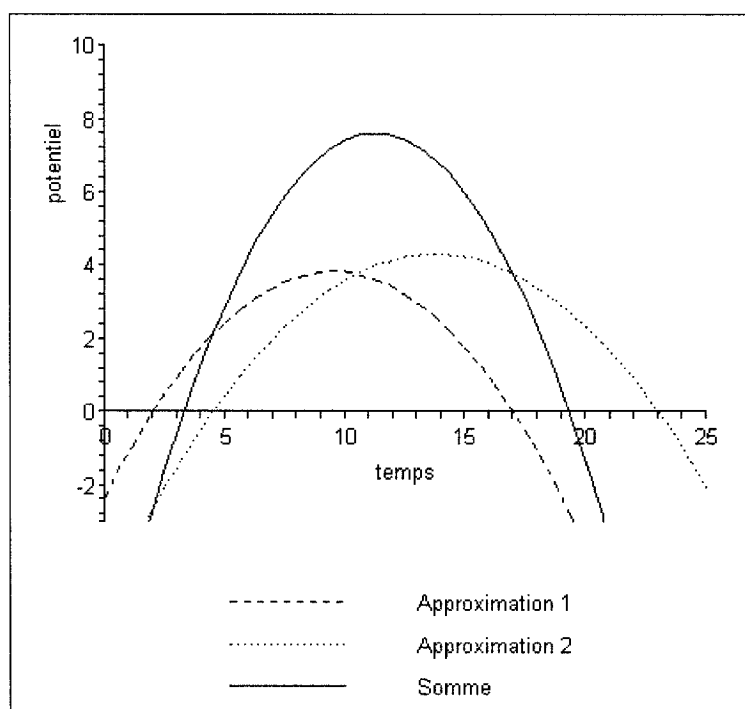


Figure 3.13 – Approximations des signaux d'entrée et de la fonction d'état

CHAPITRE 4

ALGORITHMES

D'APPRENTISSAGE

Une fois l'approximation polynomiale du quantron développée, il faut trouver un moyen de l'entraîner. Un problème d'apprentissage consiste en une instance particulière du modèle étudié (avec des valeurs cibles pour les paramètres d'apprentissage). Un algorithme d'apprentissage devrait, à partir d'un certain nombre d'exemples de ce modèle, modifier les paramètres d'une autre instance du modèle de manière à retrouver les valeurs cibles.

Pour un seul quantron, nous considérons le problème suivant : étant donné un problème de classification binaire, trouver les valeurs des paramètres \vec{w} , \vec{s} , $\vec{\theta}$ tel que le maximum de l'approximation $f(t)$ est plus élevé que le seuil Γ pour tous les exemples de la première classe, et moins élevé que Γ pour tous les exemples de la seconde classe.

Pour les approximations (3.19) et (3.48), on remarque que l'optimum global de la fonction d'état du quantron est atteint au temps suivant :

$$t_m = \frac{a_1}{2a_2}, \quad (4.1)$$

et prend la valeur suivante :

$$f_m = \frac{a_1^2}{4a_2} - a_0. \quad (4.2)$$

Si $f_m \geq \Gamma$, la sortie du quantron correspond à la plus petite racine de l'équation $f(t) = \Gamma$, qui est donnée par

$$\alpha = \frac{a_1 - \sqrt{a_1^2 - 4a_2(a_0 + \Gamma)}}{2a_2}. \quad (4.3)$$

Si a_2 est positif, l'optimum est en fait un maximum, et il dépasse le seuil si $a_1^2 - 4a_2(a_0 + \Gamma) > 0$. Si a_2 est négatif, l'optimum est un minimum, et les racines devraient être ignorées. Le modèle polynomial n'est valide que si la condition suivante est respectée :

$$0 < \alpha < t_m. \quad (4.4)$$

Si cette condition n'est pas respectée, l'approximation n'est pas valide, car le temps de premier passage du seuil est négatif. Dans ce cas, la transmission d'information

ne devrait pas avoir lieu, car la valeur de sortie du quantron ne peut pas être négative.

Nous étudierons maintenant la forme des fonctions discriminantes de l'approximation du quantron à deux entrées, et nous établirons des preuves au sujet de l'apprentissage par les fonctions discriminantes. Ensuite, nous aborderons l'apprentissage de l'approximation du quantron par la descente du gradient.

4.1 Apprentissage par les fonctions discriminantes

Nous allons maintenant montrer que la fonction discriminante de l'approximation du quantron à deux entrées est équivalente à un classificateur linéaire dans un espace de dimension plus élevé. Ce résultat nous permettra de développer un algorithme pour entraîner l'approximation du quantron.

Lemme 2. *La fonction discriminante du quantron à deux entrées avec l'approximation des PPS, où w_1 et w_2 sont positifs, est un polynôme des variables d'entrée x_1 et x_2 dont les coefficients peuvent être obtenus analytiquement.*

Preuve. Puisque a_0 et a_1 sont des fonctions de x_1 et x_2 , nous pouvons les écrire sous la forme $a_0(x_1, x_2)$ et $a_1(x_1, x_2)$. En utilisant les coefficients obtenus avec le Lemme 1, et en prenant $M = 2$, nous trouvons l'expression suivante :

$$f_m(x_1, x_2) = \frac{a_1^2(x_1, x_2)}{4a_2} - a_0(x_1, x_2). \quad (4.5)$$

Donc, il y a émission de neurotransmetteurs quand $f_m(x_1, x_2) \geq \Gamma$. La condition (3.24) nous permet d'exprimer cette inégalité de la manière suivante :

$$g(x_1, x_2) = a_1^2(x_1, x_2) - 4a_2[a_0(x_1, x_2) + \Gamma] \geq 0, \quad (4.6)$$

puisque a_2 est toujours positif quand w_1 et w_2 sont positifs. Les coefficients $a_0(x_1, x_2)$ et $a_1(x_1, x_2)$, et par conséquent la fonction discriminante $g(x_1, x_2)$ sont des polynômes en x_1 et x_2 . Nous avons :

$$\begin{aligned} a_1^2(x_1, x_2) = & \beta_1^2(N-1)^2x_1^2 + \beta_2^2(N-1)^2x_2^2 + 2\beta_1\beta_2(N-1)^2x_1x_2 \\ & + [4\beta_1^2(N-1)(\theta_1 + s_1) + 4\beta_1\beta_2(N-1)(\theta_2 + s_2)]x_1 \\ & + [4\beta_2^2(N-1)(\theta_2 + s_2) + 4\beta_1\beta_2(N-1)(\theta_1 + s_1)]x_2 \\ & + 4[\beta_1(\theta_1 + s_1) + \beta_2(\theta_2 + s_2)]^2, \end{aligned} \quad (4.7)$$

$$\begin{aligned} a_2[a_0(x_1, x_2) + \Gamma] = & \frac{(N-1)(2N-1)}{6}\beta_1(\beta_1 + \beta_2)x_1^2 + \frac{(N-1)(2N-1)}{6}\beta_2(\beta_1 + \beta_2)x_2^2 \\ & + (N-1)(\beta_1 + \beta_2)[\beta_1(\theta_1 + s_1)x_1 + \beta_2(\theta_2 + s_2)x_2] \\ & + (\beta_1 + \beta_2)[\Gamma + \beta_1(\theta_1^2 + 2\theta_1s_1) + \beta_2(\theta_2^2 + 2\theta_2s_2)]. \end{aligned} \quad (4.8)$$

et donc, nous obtenons

$$g(x_1, x_2) = e_{20}x_1^2 + e_{02}x_2^2 + e_{11}x_1x_2 + e_{10}x_1 + e_{01}x_2 + e_{00}, \quad (4.9)$$

avec

$$\begin{bmatrix} e_{00} \\ e_{01} \\ e_{02} \\ e_{10} \\ e_{11} \\ e_{20} \end{bmatrix} = \begin{bmatrix} 4\beta_1^2 s_1^2 + 4\beta_2^2 s_2^2 - 4\Gamma(\beta_1 + \beta_2) + 2\beta_1\beta_2[4s_1s_2 - 2\delta^2 + 4s_1\delta - 4s_2\delta] \\ 4(N-1)\beta_1\beta_2(s_1 - s_2 - \delta) \\ \beta_2^2(N-1)^2 - \frac{2(N-1)(2N-1)}{3}\beta_2(\beta_1 + \beta_2) \\ 4(N-1)\beta_1\beta_2(\delta + s_2 - s_1) \\ 2\beta_1\beta_2(N-1)^2 \\ \beta_1^2(N-1)^2 - \frac{2(N-1)(2N-1)}{3}\beta_1(\beta_1 + \beta_2) \end{bmatrix}, \quad (4.10)$$

où $\delta = \theta_2 - \theta_1$. Nous avons donc exprimé les coefficients de (4.9) en fonction des paramètres du quantron. \square

Nous analyserons maintenant le problème d'entraîner l'approximation du quantron par sa fonction discriminante. Nous utiliserons la terminologie de Nilsson (1965) : une Φ -machine est un séparateur linéaire pour une projection des variables d'entrée dans un espace de caractéristiques de dimension plus élevée. Habituellement, les paramètres (ou les poids) d'une Φ -machine sont des variables qui peuvent prendre n'importe quelle valeur réelle. Cependant, dans la suite, nous développons un système d'équations non linéaires sous la forme d'égalités entre les poids de la Φ -machine et

les paramètres du quantron. Ainsi, nous utilisons en fait une Φ -machine avec des contraintes sur les poids.

Lemme 3. *Le quantron à deux entrées avec l'approximation des PPS, où w_1 et w_2 sont positifs, est équivalent à une Φ -machine à quatre entrées.*

Preuve. Les relations suivantes s'appliquent aux coefficients de la fonction discriminante du Lemme 2 :

$$e_{10} = -e_{01}, \quad (4.11)$$

$$e_{20} = \frac{1 - N^2}{3} \beta_1^2 - \frac{2N - 1}{3(N - 1)} e_{11}, \quad (4.12)$$

$$e_{02} = \frac{1 - N^2}{3} \beta_2^2 - \frac{2N - 1}{3(N - 1)} e_{11}. \quad (4.13)$$

On peut modifier (4.9) pour obtenir

$$g(x_1, x_2) = \sum_{k=1}^4 c_k \Phi_k(x_1, x_2) + c_0 \quad (4.14)$$

avec

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} e_{00} \\ \beta_1^2 \\ \beta_2^2 \\ \beta_1\beta_2 \\ \beta_1\beta_2(\delta + s_2 - s_1) \end{bmatrix}, \quad (4.15)$$

et

$$\begin{bmatrix} \Phi_1(x_1, x_2) \\ \Phi_2(x_1, x_2) \\ \Phi_3(x_1, x_2) \\ \Phi_4(x_1, x_2) \end{bmatrix} = \begin{bmatrix} \frac{1-N^2}{3}x_1^2 \\ \frac{1-N^2}{3}x_2^2 \\ 2(N-1)^2x_1x_2 - \frac{2(N-1)(2N-1)}{3}(x_1^2 + x_2^2) \\ 4(N-1)(x_1 - x_2) \end{bmatrix}. \quad (4.16)$$

La fonction discriminante $g(x_1, x_2)$ est une Φ -machine, avec les entrées (x_1, x_2) du quantron projetées dans l'espace de fonctions $\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$. Les poids c_i peuvent s'exprimer en fonction des paramètres du quantron. Nous avons donc cinq contraintes qui s'appliquent sur ces poids. \square

Puisqu'une Φ -machine est un classificateur linéaire pour les entrées transformées, elle peut être entraînée avec la règle d'apprentissage du perceptron (Haykin, 1999). On obtient donc des valeurs numériques pour les poids de la Φ -machine. Il faut donc trouver les relations permettant de calculer $\beta_1, \beta_2, s_1, s_2, \delta$ à partir de c_0, c_1, c_2, c_3, c_4 .

Théorème 1. *Les paramètres du quantron à deux entrées avec l'approximation des PPS, où w_1 et w_2 sont positifs, peuvent être obtenus à partir des poids de la Φ -machine en résolvant analytiquement un problème de moindres carrés non linéaire.*

Preuve. On ne peut pas utiliser directement les coefficients c_1 et c_2 de la Φ -machine du Lemme 3 pour trouver la valeur de β_1 et β_2 , à cause de la présence de l'équation $c_3 = \beta_1\beta_2$. Les équations $c_1 = \beta_1^2$, $c_2 = \beta_2^2$ et $c_3 = \beta_1\beta_2$ forment un problème de moindres carrés non linéaire. Les valeurs de β_1 et β_2 doivent minimiser

$$h(\beta_1, \beta_2) = \frac{1}{2}(\beta_1^2 - c_1)^2 + \frac{1}{2}(\beta_2^2 - c_2)^2 + \frac{1}{2}(\beta_1\beta_2 - c_3)^2 \quad (4.17)$$

et la recherche des valeurs de β_1 et β_2 telles que $\nabla h = 0$ donne les équations suivantes :

$$2\beta_1^3 + \beta_1\beta_2^2 - 2c_1\beta_1 - c_3\beta_2 = 0 \quad (4.18)$$

$$2\beta_2^3 + \beta_1^2\beta_2 - 2c_2\beta_2 - c_3\beta_1 = 0 \quad (4.19)$$

où on peut ignorer la solution triviale $\beta_1 = \beta_2 = 0$. L'équation (4.19) est quadratique en β_1 , avec les racines suivantes :

$$\beta_1 = \frac{1}{2\beta_2} \left[c_3 \pm \sqrt{c_3^2 - 8\beta_2^4 + 8c_2\beta_2^2} \right] \quad (4.20)$$

En remplaçant β_1 par ces racines dans (4.18), nous obtenons

$$\begin{aligned} & 9\beta_2^{10} + (12c_1 - 33c_2)\beta_2^8 + (8c_3^2 + 4c_1^2 - 28c_1c_2 + 40c_2^2)\beta_2^6 \\ & + (16c_1c_2^2 - 8c_2c_3^2 - 3c_1c_3^2 - 16c_2^3 - 4c_1^2c_2)\beta_2^4 + (4c_1c_2c_3^2 - c_3^4)\beta_2^2 = 0 \end{aligned} \quad (4.21)$$

En enlevant β_2^2 de chaque terme de (4.21) (ce qui donne la racine double $\beta_2 = 0$), nous obtenons une équation quartique en β_2^2 qui peut être résolue analytiquement. Puisque la valeur minimale de (4.17) n'est pas nécessairement nulle, les valeurs trouvées de β_1 et β_2 ne donnent pas exactement la fonction discriminante de la Φ -machine entraînée. Cependant, la différence est minimale (au sens des moindres carrés).

Après avoir trouvé β_1 et β_2 , on peut procéder avec les autres paramètres. En isolant δ dans l'équation de c_4 , et en remplaçant $\beta_1\beta_2$ par c_3 , nous trouvons

$$\delta = \frac{c_4}{c_3} + s_1 - s_2, \quad (4.22)$$

que nous pouvons substituer dans l'équation de c_0 pour trouver

$$c_0 = s_1^2(4\beta_1^2 + 4\beta_1\beta_2) + s_2^2(4\beta_2^2 + 4\beta_1\beta_2) - 4\Gamma(\beta_1 + \beta_2) - \frac{4c_4^2}{c_3}. \quad (4.23)$$

Cette équation unique relie s_1 et s_2 . Nous pouvons choisir de prendre $s = s_1 = s_2$ sans perte de généralité, car il ne reste qu'une seule équation à deux inconnues. Cette réduction du nombre d'inconnues résulte en l'équation suivante

$$c_0 = 4s^2(c_1 + c_2 + 2c_3) - 4\Gamma\sqrt{c_1 + c_2 + 2c_3} - \frac{4c_4^2}{c_3}, \quad (4.24)$$

puisque $\beta_1 + \beta_2 = \sqrt{c_1 + c_2 + 2c_3}$. Nous obtenons donc

$$s = \frac{1}{2\sqrt{(c_1 + c_2 + 2c_3)}} \sqrt{c_0 + 4\Gamma\sqrt{c_1 + c_2 + 2c_3} + \frac{4c_4^2}{c_3}} \quad (4.25)$$

Avec $\beta_1, \beta_2, s_1, s_2, \delta$, il est facile de trouver les valeurs de w_1 et w_2 , et de choisir des valeurs appropriées pour θ_1 et θ_2 . Nous avons donc développé une méthode analytique pour trouver les paramètres de l'approximation du quanton à partir de c_0, c_1, c_2, c_3 et c_4 . \square

En supposant la connaissance à priori de la relation $\beta_1 = r\beta_2$, où r est une constante connue ($r > 0$), l'apprentissage des paramètres du quantron peut être fait de manière exacte. Le théorème suivant est un résultat puissant, puisqu'il démontre la possibilité d'apprendre exactement les paramètres de l'approximation du quantron. Évidemment, r n'est pas connu, mais la recherche de la bonne valeur de r peut se faire efficacement, puisqu'il n'y a qu'une seule dimension à parcourir.

Théorème 2. *Les paramètres du quantron à deux entrées avec l'approximation des PPS, dans le cas spécial où $\beta_1 = r\beta_2$, peuvent être obtenus exactement à partir des poids de la Φ -machine.*

Preuve. Nous pouvons modifier la Φ -machine du Lemme 3 ainsi :

$$g(x_1, x_2) = \omega_1 \Phi'_1(x_1, x_2) + \omega_2 \Phi'_2(x_1, x_2) + \omega_0 \quad (4.26)$$

avec

$$\begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 4(r+1)^2 \beta_1^2 s^2 - 4\Gamma(r+1)\beta_1 - 4r\beta_1^2 \delta^2 \\ \beta_1^2 \\ r\beta_1^2 \delta \end{bmatrix}, \quad (4.27)$$

et

$$\Phi'_1(x_1, x_2) = 2r(N-1)^2 x_1 x_2 - \frac{N-1}{3} [(4Nr + N - 2r + 1)x_1^2 + (Nr + 4N + r - 2)x_2^2], \quad (4.28)$$

$$\Phi'_2(x_1, x_2) = 4(N-1)(x_1 - x_2) \quad (4.29)$$

où, comme dans le Théorème 1, nous avons choisi $s = s_1 = s_2$. Par conséquent, nous obtenons

$$\beta = \sqrt{\omega_1}, \quad (4.30)$$

$$\delta = \frac{\omega_2}{r\omega_1}, \quad (4.31)$$

$$s^2 = \frac{\omega_0}{4(r+1)^2\omega_1} + \frac{\Gamma}{(r+1)\sqrt{\omega_1}} + \frac{\omega_2^2}{r(r+1)^2\omega_1^2}, \quad (4.32)$$

et nous pouvons donc exprimer les paramètres du quantron ainsi :

$$w_2 = rw_1, \quad (4.33)$$

$$w_1 = \frac{\omega_0}{4N(r+1)^2\sqrt{\omega_1}} + \frac{\Gamma}{N(r+1)} + \frac{\omega_2^2}{Nr(r+1)^2\sqrt[3]{\omega_1}}, \quad (4.34)$$

$$s = \sqrt{\frac{\omega_0}{4(r+1)^2\omega_1} + \frac{\Gamma}{(r+1)\sqrt{\omega_1}} + \frac{\omega_2^2}{r(r+1)^2\omega_1^2}}, \quad (4.35)$$

$$\theta_1 = \max \left\{ -\frac{\omega_2}{r\omega_1}, 0 \right\}, \quad (4.36)$$

$$\theta_2 = \max \left\{ \frac{\omega_2}{r\omega_1}, 0 \right\}. \quad (4.37)$$

Nous avons donc calculé les paramètres de l'approximation du quantron à partir de $\omega_0, \omega_1, \omega_2$. \square

Les deux théorèmes précédents dépendent de résultats numériques obtenus à partir de l'algorithme d'apprentissage du perceptron. Ces résultats peuvent être incohérents. Par exemple, dans le Théorème 2, s'il advenait que ω_1 soit négatif, les paramètres du quantron n'auraient pas des valeurs réelles. Si cette situation arrive, on peut appliquer de nouveau l'algorithme d'apprentissage du perceptron au problème, dans la recherche d'une nouvelle solution valide.

Pour l'approximation des signaux d'entrée du quantron, on peut également trouver une fonction discriminante sous la forme d'un polynôme en x_1 et x_2 .

Lemme 4. *La fonction discriminante du quantron à deux entrées avec l'approximation des signaux d'entrée, où w_1 et w_2 sont positifs, est un polynôme des variables d'entrée x_1 et x_2 dont les coefficients peuvent être obtenus analytiquement.*

Preuve. L'approximation de la fonction d'état du quantron est, ici aussi, un polynôme quadratique, avec les coefficients suivants :

$$\begin{bmatrix} a_0(x_1, x_2) \\ a_1(x_1, x_2) \\ a_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} \frac{k_5}{d_1} + \frac{k_6}{d_2} \\ \frac{k_3}{d_1} + \frac{k_4}{d_2} \\ \frac{k_1}{d_1} + \frac{k_2}{d_2} \end{bmatrix}, \quad (4.38)$$

où

$$\begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \end{bmatrix} = \begin{bmatrix} 2w_1x_1 + 8w_1s_1 \\ 2w_2x_2 + 8w_2s_2 \\ 2(N-1)w_1x_1^2 + 4w_1[(2N-1)s_1 + \theta_1]x_1 + 16w_1s_1(\theta_1 + s_1) \\ 2(N-1)w_2x_2^2 + 4w_2[(2N-1)s_2 + \theta_2]x_2 + 16w_2s_2(\theta_2 + s_2) \\ 2(N-1)w_1\theta_1x_1^2 + 2w_1\theta_1[2(2N-1)s_1 + \theta_1]x_1 + 8w_1s_1\theta_1(\theta_1 + 2s_1) \\ 2(N-1)w_2\theta_2x_2^2 + 2w_2\theta_2[2(2N-1)s_2 + \theta_2]x_2 + 8w_2s_2\theta_2(\theta_2 + 2s_2) \end{bmatrix}, \quad (4.39)$$

et

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} (N-1)^2x_1^3 + 4(N-1)s_1x_1^2 + 4s_1^2x_1 \\ (N-1)^2x_2^3 + 4(N-1)s_2x_2^2 + 4s_2^2x_2 \end{bmatrix}. \quad (4.40)$$

Puisque $w_1, w_2 > 0$, nous avons $a_2(x_1, x_2) > 0$ et par conséquent la fonction discriminante est donnée par

$$g(x_1, x_2) = a_1^2(x_1, x_2) - 4a_2(x_1, x_2) [a_0(x_1, x_2) + \Gamma]. \quad (4.41)$$

Le quantron émet des neurotransmetteurs si $g(x_1, x_2) \geq 0$. Cette inégalité peut être exprimée sous la forme suivante :

$$\begin{aligned} d_2^2(k_3^2 - 4k_1k_5) + 2d_1d_2(k_3k_4 - 2k_1k_6 - 2k_2k_5) \\ + d_1^2(k_4^2 - 4k_2k_6) - 4\Gamma d_1d_2(k_1d_2 + k_2d_1) \geq 0 \end{aligned} \quad (4.42)$$

Nous trouvons

$$k_3^2 - 4k_1k_5 = \sum_{m=0}^4 p_{1k} x_1^k \quad (4.43)$$

et

$$k_4^2 - 4k_2k_6 = \sum_{m=0}^4 p_{2k} x_2^k, \quad (4.44)$$

où

$$\begin{bmatrix} p_{i0} \\ p_{i1} \\ p_{i2} \\ p_{i3} \\ p_{i4} \end{bmatrix} = \begin{bmatrix} 64w_i^2 s_i^2 [s_i^2 - 6s_i\theta_i - 3\theta_i^2] \\ 32w_i^2 s_i^2 [(2N-1)s_i - 2(N-1)\theta_i] \\ 4w_i^2 [(4N^2-3)s_i^2 - 2s_i\theta_i - \theta_i^2] \\ 4(N-1)w_i^2 [(2N-1)s_i - 3\theta_i] \\ (N-1)^2 w_i^2 \end{bmatrix}, \quad (4.45)$$

et

$$k_3k_4 - 2k_1k_6 - 2k_2k_5 = \sum_{m=0}^2 \sum_{n=0}^2 b_{mn} x_1^m x_2^n \quad (4.46)$$

où

$$\begin{bmatrix} b_{00} \\ b_{01} \\ b_{02} \\ b_{10} \\ b_{11} \\ b_{12} \\ b_{20} \\ b_{21} \\ b_{22} \end{bmatrix} = \begin{bmatrix} 64w_1w_2s_1s_2(m_1 + \theta_1\theta_2 - 2\theta_1^2 - 2\theta_2^2) \\ 16w_1w_2[m_2 - s_1(\theta_2^2 + 2\theta_1^2) + s_1s_2\theta_1\theta_2] \\ 8(N-1)w_1w_2s_1(s_1 + \theta_1 - 2\theta_2) \\ 16w_1w_2[m_3 - s_2(\theta_1^2 + 2\theta_2^2) + s_1s_2\theta_1\theta_2] \\ 4w_1w_2[(2N-1)m_4 + \theta_1\theta_2 - \theta_1^2 - \theta_2^2] \\ 2(N-1)w_1w_2[(2N-1)s_1 + \theta_1 - 2\theta_2] \\ 8(N-1)w_1w_2s_2(s_2 + \theta_2 - 2\theta_1) \\ 2(N-1)w_1w_2[(2N-1)s_2 + \theta_2 - 2\theta_1] \\ (N-1)^2w_1w_2 \end{bmatrix} \quad (4.47)$$

et

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \begin{bmatrix} s_1s_2 + s_1\theta_2 + s_2\theta_1 - 4s_1\theta_1 - 4s_2\theta_2 \\ (2N-1)s_1s_2(s_1 + \theta_1 - 2\theta_2) + s_1^2(\theta_2 - 4\theta_1) \\ (2N-1)s_1s_2(s_2 + \theta_2 - 2\theta_1) + s_2^2(\theta_1 - 4\theta_2) \\ s_1\theta_2 + s_2\theta_1 - 2s_1\theta_1 - 2s_2\theta_2 + (2N-1)s_1s_2 \end{bmatrix}. \quad (4.48)$$

En notant $k_i = k_{i1}x_i + k_{i0}$ et $d_i = d_{i3}x_i^3 + d_{i2}x_i^2 + d_{i1}x_i$ for $i = 1, 2$, nous trouvons l'expression suivante pour $g(x_1, x_2)$:

$$\begin{aligned}
g(x_1, x_2) &= \sum_{k=1}^3 \sum_{l=1}^3 \sum_{m=0}^4 (d_{2k}d_{2l}p_{1m}x_1^m x_2^{k+l} + d_{1k}d_{1l}p_{2m}x_1^{k+l} x_2^m) \\
&+ \sum_{k=1}^3 \sum_{l=1}^3 \sum_{m=0}^2 \sum_{n=0}^2 d_{1k}d_{2l}b_{mn}x_1^{k+m} x_2^{l+n} \\
&- 4\Gamma \sum_{k=1}^3 \sum_{l=1}^3 \sum_{m=1}^3 \sum_{n=0}^1 (d_{1k}d_{2l}d_{2m}k_{1n}x_1^{k+n} x_2^{l+m} + d_{1l}d_{1m}d_{2k}k_{2n}x_1^{l+m} x_2^{k+n})
\end{aligned} \tag{4.49}$$

Finalement, nous avons

$$g(x_1, x_2) = \sum_{i=1}^5 \sum_{j=1}^5 e_{ij} x_1^i x_2^j + \sum_{k=2}^6 (e_{k0} x_1^k + e_{0k} x_2^k) + \sum_{k=1}^4 (e_{6k} x_1^6 x_2^k + e_{k6} x_1^k x_2^6). \tag{4.50}$$

Déterminer les coefficients e_{ij} consiste tout simplement à regrouper les termes de (4.49). Ceci peut se faire avec un logiciel comme **Maple**. Les valeurs des coefficients sont présentés dans l'Annexe A. \square

Le calcul des paramètres de l'approximation du quantron à partir des valeurs de e_{ij} est beaucoup plus complexe que dans le cas précédent. Il faudrait résoudre un système qui possède beaucoup plus d'équations que de variables. Ceci ne peut être effectué de manière exacte, comme dans le Théorème 2.

4.2 Expérience 1 : Fonctions discriminantes

Nous présentons maintenant deux expériences effectuées avec les fonctions discriminantes développées précédemment. D'abord, nous appliquons le Théorème 1

à un exemple de problème d'apprentissage, puis nous évaluons la performance des approximations dans la reproduction des fonctions discriminantes du quantron.

4.2.1 Expérience 1A : Application du théorème 1

Nous présentons d'abord un exemple de l'application du théorème 1 à un quantron où $N = 8$ et $\Gamma = 10$, et dont les paramètres sont $w_1^* = 4.25$, $w_2^* = 1.5$, $s_1^* = 4$, $s_2^* = 2$, $\theta_1^* = 0$ et $\theta_2^* = 1$. L'ensemble d'apprentissage est constitué de 200 points choisis au hasard dans le rectangle $[0.5, 1.49]^2$. Nous devrions pouvoir retrouver, avec une certaine erreur, les valeurs de $w_1^*, w_2^*, s_1^*, s_2^*, \theta_1^* = 0, \theta_2^* = 0$ en utilisant le Théorème 1. La figure 4.1 présente la surface discriminante recherchée, et la figure 4.2 montre les points composant l'ensemble d'apprentissage.

Avec une application de l'algorithme d'apprentissage du perceptron, nous avons trouvé $c_0 = 98.0100$, $c_1 = 0.1863$, $c_2 = 1.5626$, $c_3 = 0.8066$ et $c_4 = -1.8029$. Puisqu'à chaque itération de l'algorithme un exemple de l'ensemble d'apprentissage est choisi au hasard pour modifier les paramètres, il est évident que ces résultats numériques vont varier sur différents essais de l'algorithme. L'implémentation de l'algorithme d'apprentissage du perceptron (en **Matlab**) est donnée dans l'Annexe B.

Le problème des moindres carrés a été résolu analytiquement à l'aide du logiciel **Maple**, avec comme solution $(\beta_1, \beta_2) = (0.5551, 1.2595)$ (les valeurs numériques sont tronquées). Les autres équations développées dans le Théorème 1 ont été utilisées pour trouver les valeurs suivantes : $w_1 = 0.9673$, $w_2 = 2.1948$, $s_1 = 3.7337$, $s_2 = 3.7337$, $\theta_1 = 2.2352$, $\theta_2 = 0$.

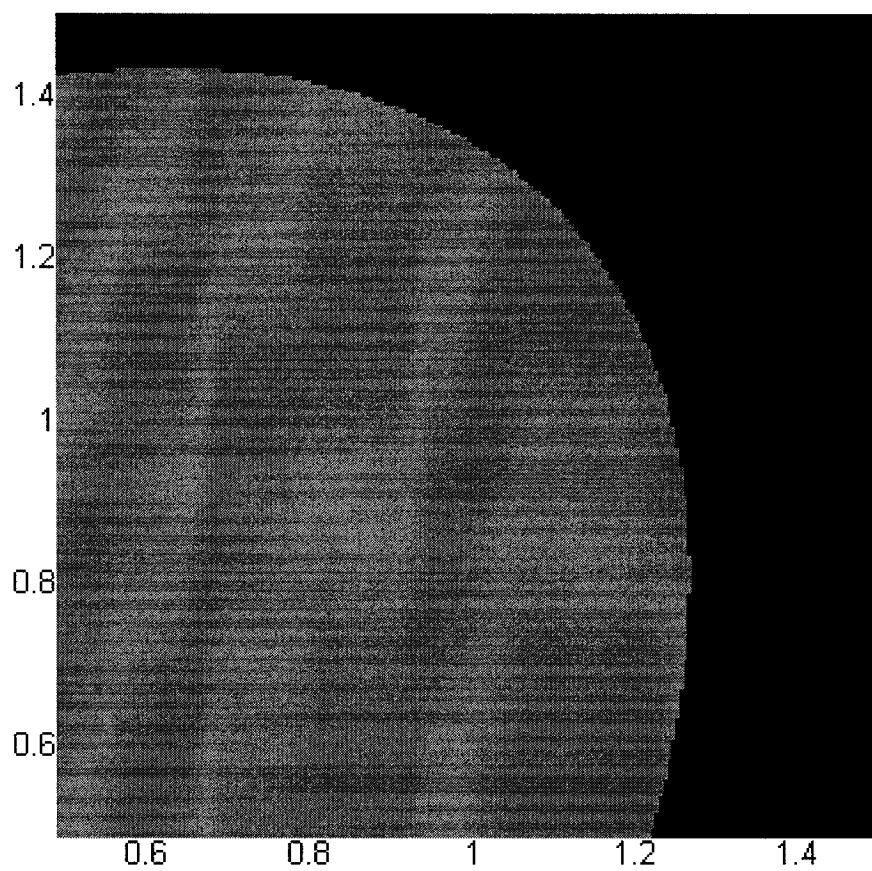


Figure 4.1 – Surface discriminante de l'expérience 1A

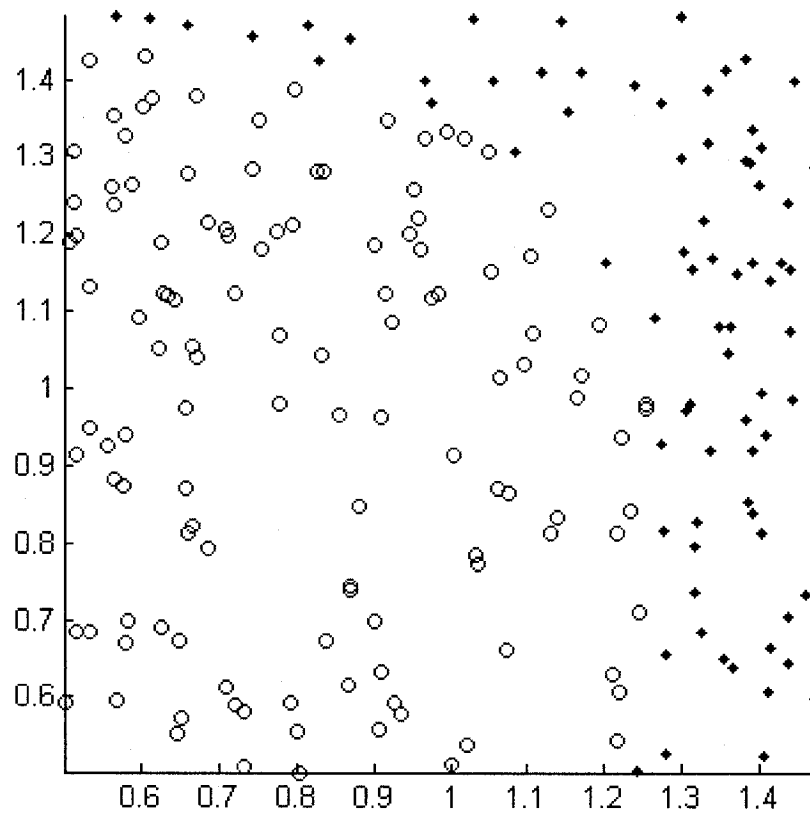


Figure 4.2 – Ensemble d'apprentissage de l'expérience 1A

La figure 4.3 montre l'application de la surface discriminante trouvée avec le théorème 1 sur l'ensemble d'entraînement. Les erreurs sont représentées par des croix, et la région où ces erreurs ont lieu est mise en évidence. L'erreur sur l'ensemble d'apprentissage est donc de 2%.

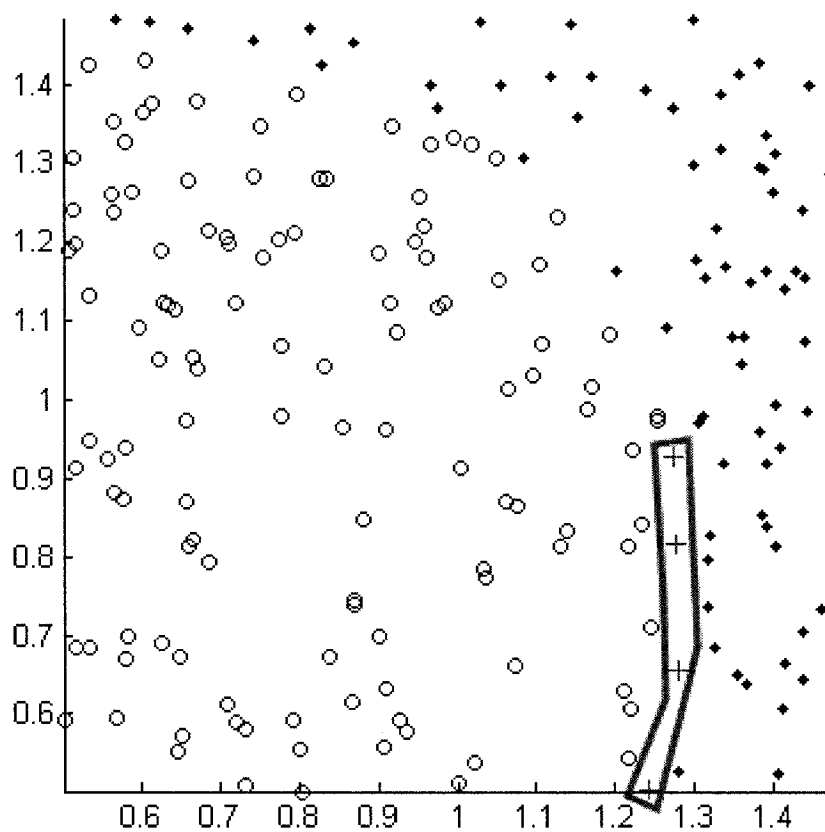


Figure 4.3 – Surface discriminante résultante

Ce résultat confirme que la résolution par la méthode des moindres carrés ne garantit pas un apprentissage exact de l'ensemble d'entraînement. Nous voyons par contre que la fonction discriminante trouvée est très proche de la solution optimale.

4.2.2 Expérience 1B : Étude comparative des fonctions discriminantes

Nous présentons maintenant une expérimentation par comparaison d'images pour analyser la performance des deux approximations dans leur reproduction des fonctions discriminantes du modèle mathématique du quantron (le modèle avec PPS rectangulaires est utilisé). Puisque notre analyse de l'apprentissage des approximations polynomiales du quantron s'est restreinte au cas où w_1 et w_2 sont positifs, nous ferons de même ici. Dans les images suivantes, les sections identifiées par le chiffre 1 indiquent que le quantron émet des neurotransmetteurs quand ses entrées s'y trouvent, tandis que le quantron n'émet rien pour les entrées situées dans les sections identifiées par le chiffre 2. Bien que la comparaison de surfaces discriminantes ne consiste pas en une méthode d'apprentissage, il est important de vérifier les capacités de reproduction des approximations, pour savoir dans quelle mesure les théorèmes développés pourront être utilisés pour guider l'apprentissage d'un quantron.

La figure 4.4 présente des exemples de fonctions discriminantes d'un quantron à deux entrées avec la première approximation. Nous reconnaissons des formes elliptiques, ce qui est normal puisque l'ensemble des zéros de (4.9) constitue une conique (l'étude des paramètres de l'équation permet de vérifier qu'il s'agit de surfaces elliptiques). Ces surfaces discriminantes sont peu ressemblantes à celles du modèle original du quantron. La figure 4.5 présente la surface discriminante originale du quantron, ainsi que son approximation par la première méthode. En général, nous remarquons que la différence entre les surfaces discriminantes, dans le cas de la première approximation, est trop grande pour utiliser directement les paramètres de l'approximation

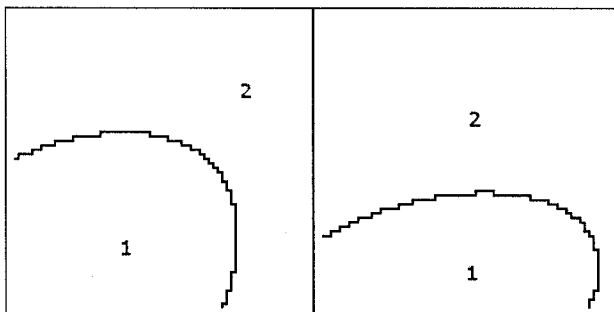


Figure 4.4 – Surfaces discriminantes de la première approximation

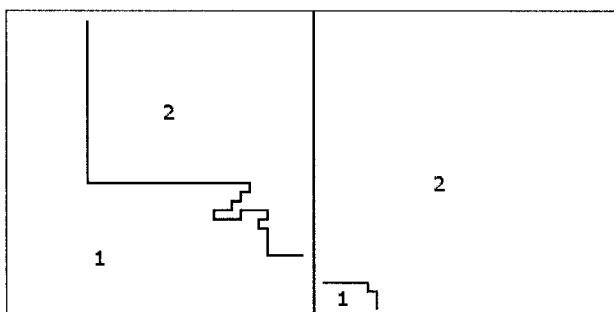


Figure 4.5 – Surface discriminante du quantron (gauche) et son approximation par la première méthode (droite)

dans le modèle original du quantron. Cette différence provient de l'erreur de modélisation qui est causée par les parties négatives des fonctions quadratiques utilisées pour approximer les PPS. Ainsi, l'intérêt de l'approximation des PPS se situe dans son expression analytique plus simple, ainsi que ses théorèmes de convergence, mais il serait étonnant qu'un algorithme d'apprentissage basé sur cette approximation soit très utile pour des réseaux de quantrons.

Les figures 4.6 et 4.7 présentent la comparaison entre six exemples de surfaces discriminantes du quantron à deux entrées et de son approximation par la deuxième méthode.

Nous avons mesuré l'erreur de reproduction des surfaces discriminantes (en pourcentage de pixels mal reproduits) sur 50 ensembles aléatoires de paramètres. La dimension des images de test est de 32×32 pixels. Nous avons obtenu une erreur moyenne de 15.27% avec un écart-type de 17.65%. La figure 4.8 présente un histogramme de l'erreur. Les deux plus petites erreurs mesurées sont 0.29% et 0.39%, tandis que les deux plus grandes erreurs sont 67.29% et 85.55%.

Les résultats de l'approximation des signaux laissent croire que cette méthode devrait être plus utile pour le développement d'un algorithme d'apprentissage pour le quanton que l'approximation des PPS. La performance de cette approche s'explique par l'approximation des signaux d'entrée en tenant compte de l'interaction de plusieurs PPS, et non de l'approximation des PPS pris individuellement. Par contre, l'analyse des fonctions discriminantes s'applique difficilement à cette approximation. Nous chercherons donc à développer une méthode d'optimisation par descente du gradient pour pallier à cette limitation.

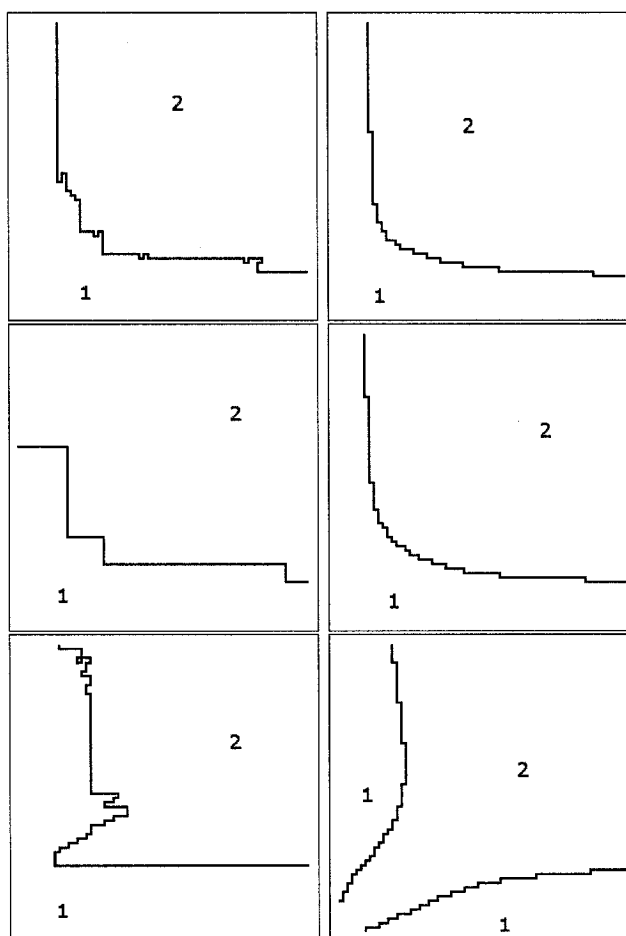


Figure 4.6 – Surfaces discriminantes du quantron (colonne de gauche) et leur approximation par la deuxième méthode (colonne de droite). Erreurs : 2.81% (haut), 10.97% (centre), 12.70% (bas)

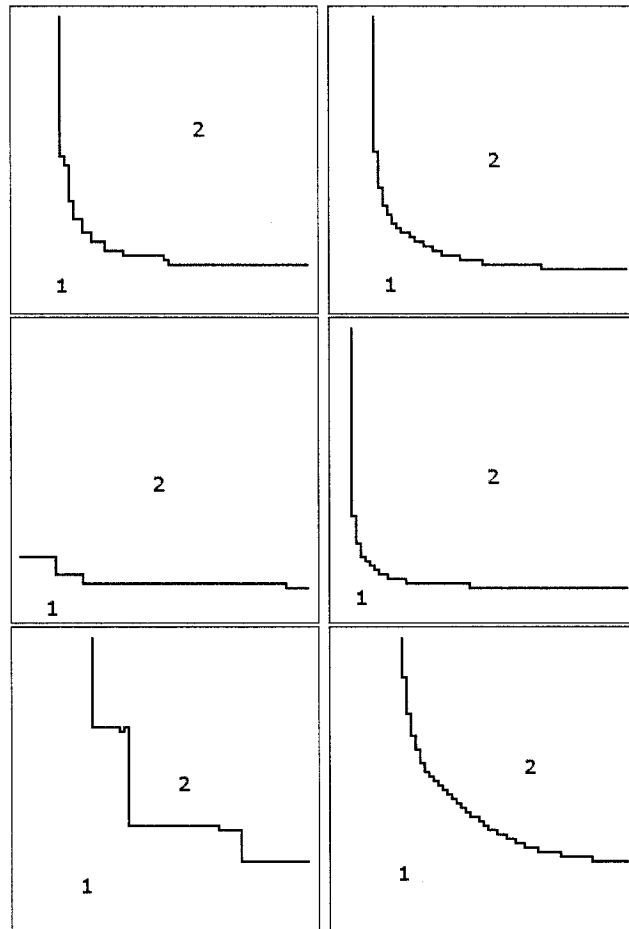


Figure 4.7 – Surfaces discriminantes du quantron (colonne de gauche) et leur approximation par la deuxième méthode (colonne de droite). Erreurs : 1.95% (haut), 4.96% (centre), 4.81% (bas)

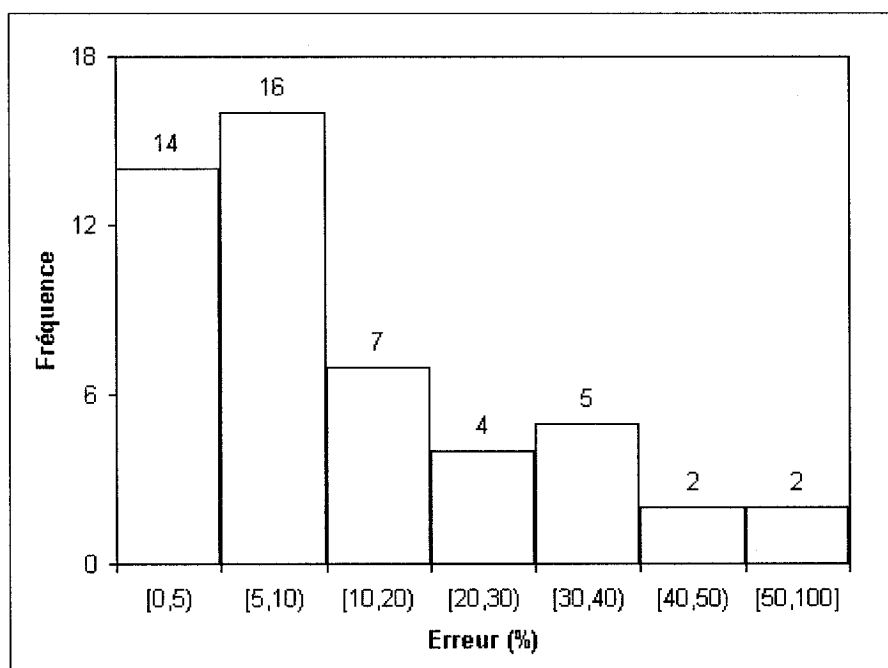


Figure 4.8 – Histogramme des erreurs de reproduction des surfaces discriminantes

4.3 Apprentissage par la descente du gradient

Puisqu'il est difficile d'entraîner directement la surface discriminante dans le cas de l'approximation des signaux, nous allons maintenant présenter comment effectuer l'apprentissage du quantron par la descente du gradient. Pour faire cela, il faut définir une fonction d'erreur qui exprime la différence entre le résultat espéré et le résultat obtenu pour un exemple de l'ensemble d'entraînement.

Pour exprimer l'erreur du quantron sur un ensemble d'entraînement de n exemples, nous proposons la fonction d'erreur suivante :

$$E = \sum_{k=1}^n (d_k e^{-c(f_m(\vec{x}_k) - \Gamma)} + (1 - d_k) e^{c(f_m(\vec{x}_k) - \Gamma)}) \quad (4.51)$$

où d_k est la classe désirée (1 ou 0) de la sortie du quantron pour l'exemple \vec{x}_k . La constante c pourrait être n'importe quel nombre réel positif. Si c est négatif, E ne mesure plus une erreur, car les exemples bien classés y contribuent plus que les exemples mal classés. Si c est nul, $E = n$ indépendamment de la classification des exemples.

La présence des termes exponentiels dans la fonction d'erreur permet d'utiliser directement la différence entre f_m et Γ comme mesure d'erreur. Ainsi, si f_m n'est pas du bon côté de Γ , l'erreur est appréciable, tandis que si f_m est du bon côté de Γ , l'erreur est négligeable.

Les paramètres du quantron sont modifiés de manière à suivre le gradient négatif de la fonction d'erreur. Soit p un paramètre du quantron. La règle de modification par descente du gradient est :

$$p' = p - \eta \frac{\partial E}{\partial p} \quad (4.52)$$

où η est le taux d'apprentissage. Nous trouvons

$$\frac{\partial E}{\partial p} = -c \sum_{k=1}^n (d_k e^{-c(f_m(\bar{x}_k) - \Gamma)} - (1 - d_k) e^{c(f_m(\bar{x}_k) - \Gamma)}) \frac{\partial f_m}{\partial p}, \quad (4.53)$$

où

$$\frac{\partial f_m}{\partial p} = \frac{a_1}{2a_2} \frac{\partial a_1}{\partial p} - \left(\frac{a_1}{2a_2} \right)^2 \frac{\partial a_2}{\partial p} - \frac{\partial a_0}{\partial p}. \quad (4.54)$$

où les coefficients a_0 , a_1 et a_2 proviennent de l'équation (3.49). Les dérivées partielles des coefficients de l'approximation polynomiale de la fonction d'état, dans le cas de l'approximation des signaux, sont données par :

$$\begin{bmatrix} \frac{\partial a_0}{\partial w_1} \\ \frac{\partial a_0}{\partial w_2} \\ \frac{\partial a_0}{\partial s_1} \\ \frac{\partial a_0}{\partial s_2} \\ \frac{\partial a_0}{\partial \theta_1} \\ \frac{\partial a_0}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} \frac{2(4s_1+x_1)[\theta_1^2+(N-1)\theta_1x_1+2\theta_1s_1]}{x_1[(N-1)x_1+2s_1]^2} \\ \frac{2(4s_2+x_2)[\theta_2^2+(N-1)\theta_2x_2+2\theta_2s_2]}{x_2[(N-1)x_2+2s_2]^2} \\ \frac{-4w_1\theta_1[-2(N-2)\theta_1x_1+4\theta_1s_1-(N-1)(2N-3)x_1^2-2(2N-3)x_1s_1]}{x_1[(N-1)x_1+2s_1]^3} \\ \frac{-4w_2\theta_2[-2(N-2)\theta_2x_2+4\theta_2s_2-(N-1)(2N-3)x_2^2-2(2N-3)x_2s_2]}{x_2[(N-1)x_2+2s_2]^3} \\ \frac{2w_1(4s_1+x_1)(2\theta_1+(N-1)x_1+2s_1)}{x_1[(N-1)x_1+2s_1]^2} \\ \frac{2w_2(4s_2+x_2)(2\theta_2+(N-1)x_2+2s_2)}{x_2[(N-1)x_2+2s_2]^2} \end{bmatrix}, \quad (4.55)$$

$$\begin{bmatrix} \frac{\partial a_1}{\partial w_1} \\ \frac{\partial a_1}{\partial w_2} \\ \frac{\partial a_1}{\partial s_1} \\ \frac{\partial a_1}{\partial s_2} \\ \frac{\partial a_1}{\partial \theta_1} \\ \frac{\partial a_1}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} \frac{2(4s_1+x_1)[2\theta_1+(N-1)x_1+2s_1]}{x_1[(N-1)x_1+2s_1]^2} \\ \frac{2(4s_2+x_2)[2\theta_2+(N-1)x_2+2s_2]}{x_2[(N-1)x_2+2s_2]^2} \\ \frac{-4w_1[-4(N-2)\theta_1x_1+8\theta_1s_1-(N-1)(2N-3)x_1^2-2(2N-3)x_1s_1]}{x_1[(N-1)x_1+2s_1]^3} \\ \frac{-4w_2[-4(N-2)\theta_2x_2+8\theta_2s_2-(N-1)(2N-3)x_2^2-2(2N-3)x_2s_2]}{x_2[(N-1)x_2+2s_2]^3} \\ \frac{4w_1(4s_1+x_1)}{x_1[(N-1)x_1+2s_1]^2} \\ \frac{4w_2(4s_2+x_2)}{x_2[(N-1)x_2+2s_2]^2} \end{bmatrix}, \quad (4.56)$$

et

$$\begin{bmatrix} \frac{\partial a_2}{\partial w_1} \\ \frac{\partial a_2}{\partial w_2} \\ \frac{\partial a_2}{\partial s_1} \\ \frac{\partial a_2}{\partial s_2} \\ \frac{\partial a_2}{\partial \theta_1} \\ \frac{\partial a_2}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} \frac{2(4s_1+x_1)}{x_1[(N-1)x_1+2s_1]^2} \\ \frac{2(4s_2+x_2)}{x_2[(N-1)x_2+2s_2]^2} \\ \frac{-8w_1[2s_1-(N-2)x_1]}{x_1[(N-1)x_1+2s_1]^3} \\ \frac{-8w_2[2s_2-(N-2)x_2]}{x_2[(N-1)x_2+2s_2]^3} \\ 0 \\ 0 \end{bmatrix}. \quad (4.57)$$

4.4 Expérience 2 : Descente du gradient

Nous présentons maintenant l'application de la descente du gradient à l'approximation des signaux d'entrée du quantron.

4.4.1 Détails d'implémentation

L'implémentation de la descente du gradient pour le quantron nécessite quelques précisions. L'utilisation directe des équations pose problème, car il est possible, pour certaines valeurs des paramètres, que a_2 soit très proche de 0, et alors le calcul de $\frac{\partial f_m}{\partial p}$ sera numériquement instable. En imposant, comme dans l'apprentissage par les fonctions discriminantes, la condition que w_1 et w_2 soient positifs, le problème peut être évité.

Nous avons intégré une recherche linéaire par *backtracking* (Nocedal et Wright, 1999) à l'algorithme de descente du gradient pour trouver un taux d'apprentissage qui assure une diminution de la valeur de E à chaque itération. Le *backtracking* consiste en une recherche itérative d'une valeur de η , dans l'équation (4.52), qui soit telle que $E(w'_1, w'_2, s'_1, s'_2, \theta'_1, \theta'_2) < E(w_1, w_2, s_1, s_2, \theta_1, \theta_2)$.

4.4.2 Résultats

Nous avons généré un problème d'apprentissage aléatoire de 500 points dont les exemples sont situés dans le rectangle $[1.0, 2.5]^2$, avec un quantron où $N = 8$ et $\Gamma = 10$, et dont les paramètres sont $w_1^* = 0.2578$, $w_2^* = 3.9611$, $s_1^* = 9.4965$, $s_2^* = 0.8877$, $\theta_1^* = 1.7637$ et $\theta_2^* = 3.8850$.

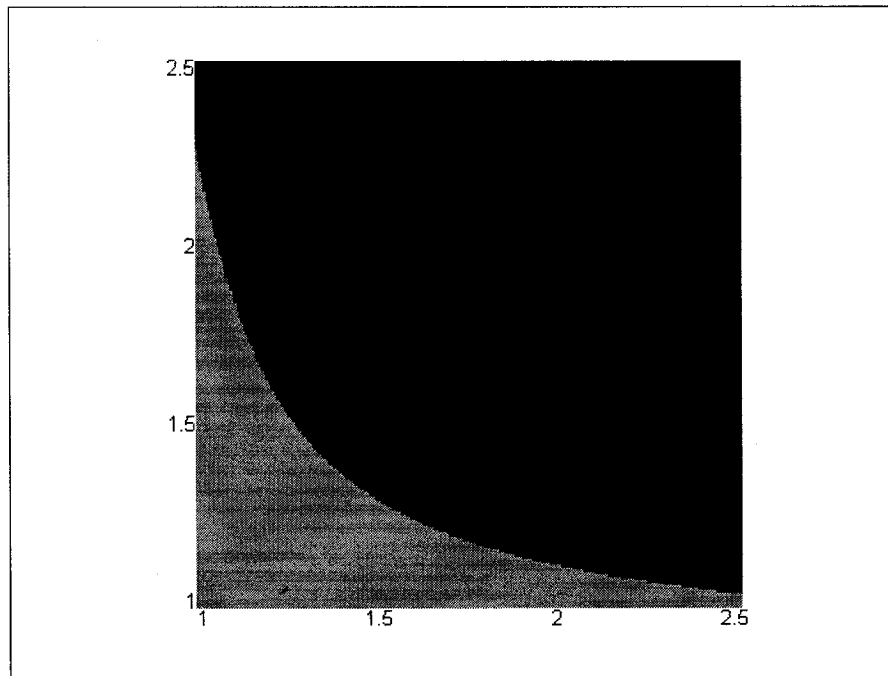


Figure 4.9 – Surface discriminante de l'expérience 2

Les figures 4.9 et 4.10 illustrent respectivement la surface discriminante et le problème d'apprentissage considéré. La couleur grise correspond aux entrées pour lesquelles le quantron émet des neurotransmetteurs, et la couleur noire représente celles où rien n'est émis.

L'apprentissage a été effectué avec 10 points de départ différents choisis au hasard avec $w_1, w_2 \in (0, 6)$ et $s_1, s_2, \theta_1, \theta_2 \in (0, 10)$ pendant 100 itérations, pour $c = 1, 2, 4, 6, 8, 10, 12$. Les données recueillies sont E_{100} (la valeur de E après 100 itérations), ν_{100} (le nombre d'exemples mal classés après 100 itérations), et ν_m (le nombre minimal d'exemples mal classés rencontré pendant les 100 itérations).

Le tableau 4.1 présente les valeurs moyennes de $E_{100}, \nu_{100}, \nu_m$ pour chaque valeur

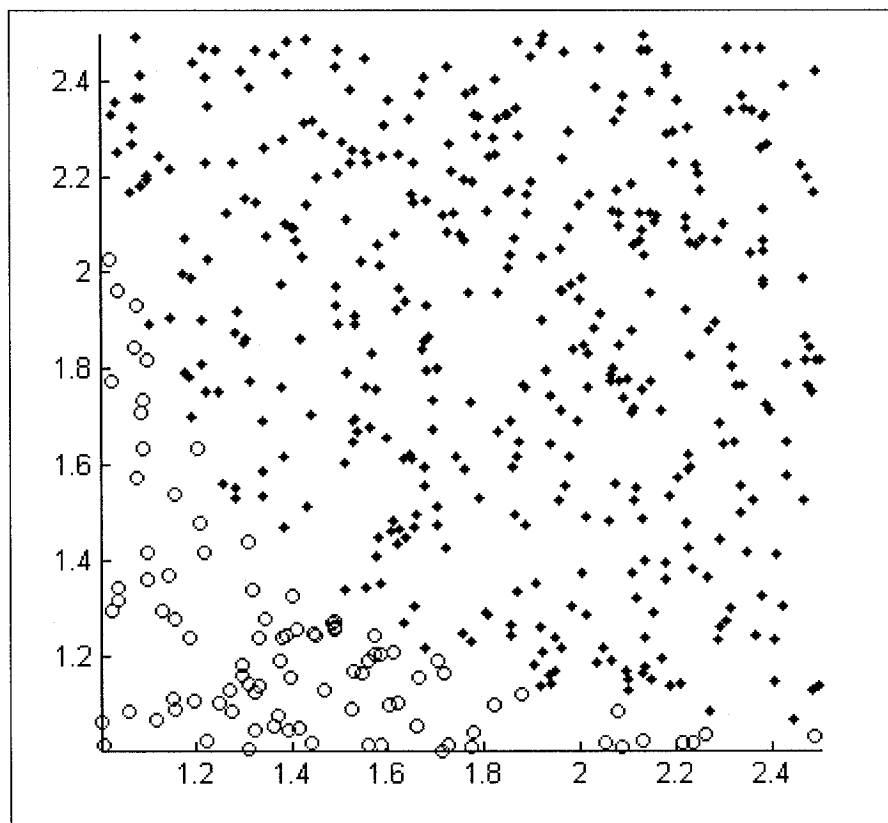


Figure 4.10 – Ensemble d'apprentissage de l'expérience 2

Tableau 4.1 – Résultats moyens pour différentes valeurs de c

| c | E_{100} | ν_{100} | $\nu_{100}(\%)$ | ν_m | $\nu_m(\%)$ |
|-----|-----------|-------------|-----------------|---------|-------------|
| 1 | 165.0006 | 33.8 | 6.76% | 24.7 | 4.94% |
| 2 | 98.9191 | 15.4 | 3.08% | 12.8 | 2.56% |
| 4 | 55.3652 | 12.8 | 2.56% | 11 | 2.20% |
| 6 | 45.5115 | 10.0 | 2.00% | 8.7 | 1.74% |
| 8 | 64.4263 | 12.2 | 2.44% | 10.6 | 2.12% |
| 10 | 94.0123 | 15.3 | 3.06% | 13.5 | 2.70% |
| 12 | 100.5356 | 13.8 | 2.76% | 12.1 | 2.42% |

de c . Les résultats détaillés sont présentés dans l'Annexe C.

Il faut faire attention à l'interprétation de E_{100} . Quand la valeur de c change, la valeur ajoutée à E pour chaque exemple mal classé sera beaucoup plus grande si c est grand, mais la valeur ajoutée à E pour chaque exemple bien classé sera plus petite.

L'étude du comportement de l'algorithme en cours d'apprentissage permet de remarquer qu'il est possible que le nombre d'exemples mal classés augmente lorsque E baisse. La figure 4.11 illustre un exemple d'application de l'algorithme d'apprentissage (avec $c = 8$) où on peut observer ce phénomène.

L'utilisation de valeurs de c entre 0 et 1 ne donne pas de bons résultats. De plus, il est possible que le point de départ aléatoire cause des problèmes d'évaluation numérique de la fonction E . Par exemple, un exemple mal classé avec $|f_m - \Gamma| = 50$ et $c = 20$ aurait un terme d'erreur de e^{1000} , ce qui dépasse le nombre maximal défini

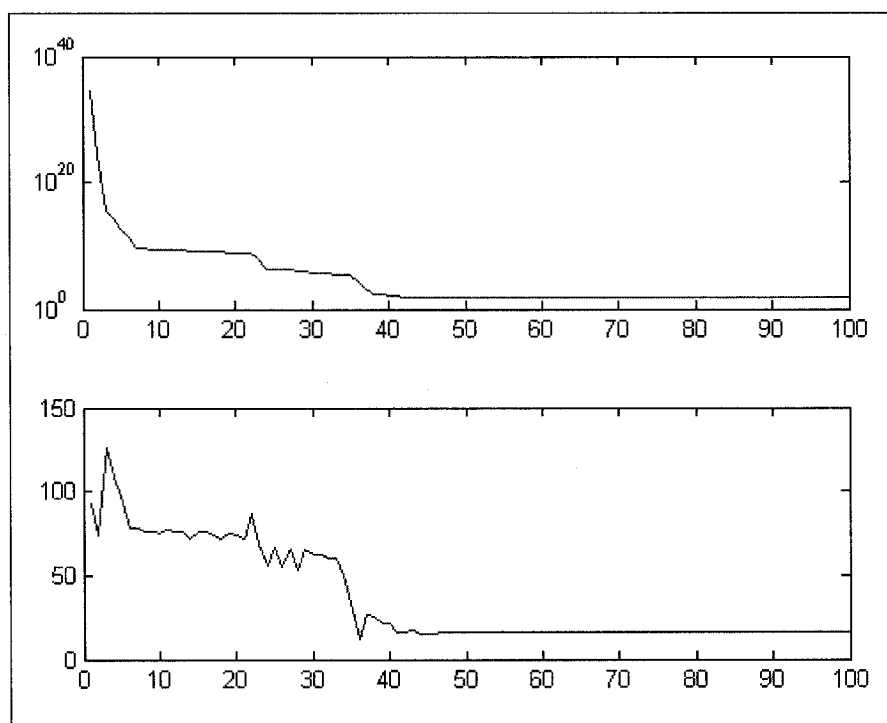


Figure 4.11 – Courbes de E (haut) et ν (bas) sur 100 itérations

dans la norme IEEE-754 (environ $1.8 \times 10^{+308}$). Il serait donc utile de développer une heuristique simple permettant de trouver des paramètres de départ pour lesquels l'erreur n'est pas trop grande.

Les résultats obtenus montrent qu'avec un choix de c entre 2 et 12, l'apprentissage se fait, en moyenne, avec une erreur assez faible (entre 2.00% et 3.08%).

CHAPITRE 5

RÉSEAUX DE QUANTRONS

Tout comme les autres types de neurones, il est possible de combiner plusieurs quantrons pour former un réseau. Beaucoup plus puissant qu'un simple quantron, un réseau de quantrons permet de produire des surfaces discriminantes d'une grande variété. Cependant, l'apprentissage avec un tel réseau fait face à une nouvelle difficulté. En effet, il est possible que des neurones (ou des groupes de neurones) du réseau soient "silencieux", c'est-à-dire qu'ils n'émettront jamais de neurotransmetteurs. Cette situation peut survenir, par exemple, si les valeurs des poids d'un neurone sont beaucoup trop faibles pour que le seuil soit atteint. La présence de neurones silencieux complique l'adaptation de rétro-propagation aux réseaux de quantrons.

5.1 Neurones silencieux

La principale difficulté avec le calcul direct du gradient d'une fonction d'erreur pour le quantron est la nature de sa sortie. Le gradient de la sortie par rapport aux paramètres n'est défini que si des neurotransmetteurs sont émis, puisque le temps de premier passage du seuil n'existe pas si le quantron n'émet rien. S'il advenait que pour chaque exemple d'un ensemble d'entraînement un quantron n'émettait rien, l'apprentissage serait impossible, car les paramètres ne pourraient pas varier. On peut vérifier ce phénomène mathématiquement.

Soit le réseau de trois quantrons de la figure 5.1. La sortie q_k (pour $k = 1, 2$) est

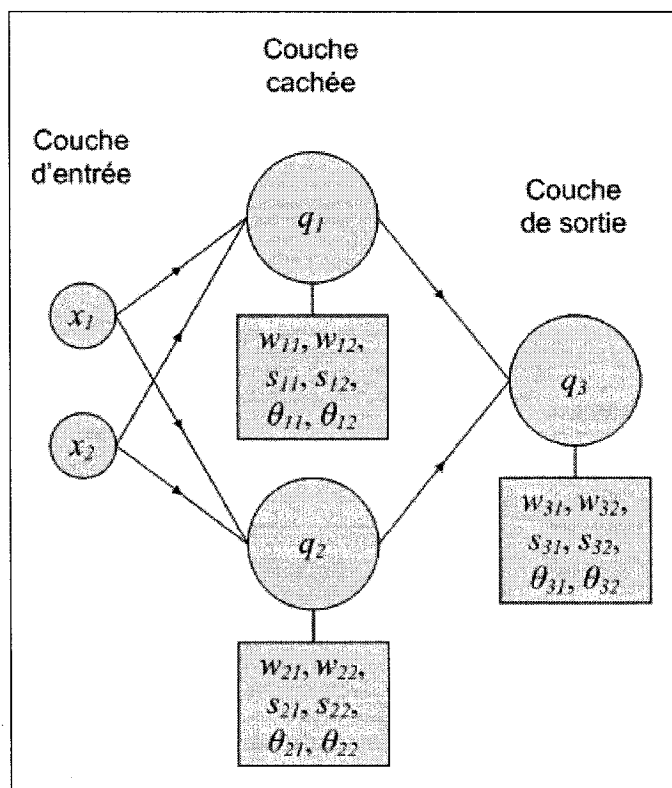


Figure 5.1 – Réseau de trois quantons

définie comme suit :

$$q_k(x_1, x_2) = \begin{cases} \alpha(x_1, x_2, w_{k1}, w_{k2}, s_{k1}, s_{k2}, \theta_{k1}, \theta_{k2}) & \text{si } f_k(x_1, x_2) \geq \Gamma \\ \perp & \text{si } f_k(x_1, x_2) < \Gamma \end{cases}, \quad (5.1)$$

où α est le temps de dépassement du seuil, et où \perp est une constante symbolique qui indique que le quantron n'émet pas de neurotransmetteurs. La sortie q_3 est définie comme suit :

$$q_3(x_1, x_2) = \begin{cases} 1 & \text{si } f_3(q_1, q_2) \geq \Gamma \\ 0 & \text{si } f_3(q_1, q_2) < \Gamma \end{cases}. \quad (5.2)$$

Ici, f_k désigne l'application de f_m au neurone k . Nous avons donc

$$\begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \\ f_3(q_1, q_2) \end{bmatrix} = \begin{bmatrix} f_m(x_1, x_2, w_{11}, w_{12}, s_{11}, s_{12}, \theta_{11}, \theta_{12}) \\ f_m(x_1, x_2, w_{21}, w_{22}, s_{21}, s_{22}, \theta_{21}, \theta_{22}) \\ f_m(q_1(x_1, x_2), q_2(x_1, x_2), w_{31}, w_{32}, s_{31}, s_{32}, \theta_{31}, \theta_{32}) \end{bmatrix}. \quad (5.3)$$

De plus, la nature de \perp impose les équivalences suivantes à f_3 :

$$f_3(\alpha, \perp) \equiv f_m^*(q_1(x_1, x_2), w_{31}, s_{31}, \theta_{31}), \quad (5.4)$$

$$f_3(\perp, \alpha) \equiv f_m^*(q_2(x_1, x_2), w_{32}, s_{32}, \theta_{32}), \quad (5.5)$$

$$f_3(\perp, \perp) \equiv 0, \quad (5.6)$$

où f_m^* est la restriction de f_m à seulement une entrée.

Maintenant, il est possible de vérifier que l'apprentissage d'un neurone silencieux par la descente du gradient est impossible. Nous supposons que le neurone 1 est silencieux, c'est-à-dire que $q_1(x_1, x_2) = \perp$ pour tous les exemples de l'ensemble d'entraînement. Nous notons d la valeur désirée du réseau pour les entrées x_1 et x_2 , et nous exprimons l'erreur du réseau ainsi :

$$E = f_e(x_1, x_2, d, f_3) \quad (5.7)$$

où f_e est une fonction d'erreur. Nous remarquons qu'il s'agit de l'erreur instantanée sur la présentation d'un élément quelconque de l'ensemble d'entraînement. Prenons, par exemple, la règle de modification suivante pour le paramètre w_{11} :

$$w_{11}(n+1) = w_{11}(n) - \eta \frac{\partial E}{\partial w_{11}}, \quad (5.8)$$

où η est le taux d'apprentissage. Le calcul de la dérivée partielle de la fonction d'erreur s'effectue par la règle de dérivation en chaîne :

$$\frac{\partial E}{\partial w_{11}} = \frac{\partial f_e(x_1, x_2, d, f_3)}{\partial f_3} \frac{\partial f_3}{\partial w_{11}}. \quad (5.9)$$

Puisque $q_1(x_1, x_2) = \perp$, nous avons

$$\frac{\partial f_3}{\partial w_{11}} = \frac{\partial f_m^*(q_2(x_1, x_2), w_{32}, s_{32}, \theta_{32})}{\partial w_{11}} = 0. \quad (5.10)$$

Comme le neurone 1 est silencieux, ce calcul est valide pour n'importe quel élément de l'ensemble d'entraînement, et w_{11} ne sera jamais modifié. Il en est de même pour les autres paramètres du neurone 1. Or, si aucun des paramètres du neurone 1 ne peut être modifié, il restera toujours silencieux.

Dans l'étude de *Spikeprop*, ce problème survient également (Bothe, Kok et La Poutré, 2002 ; Schrauwen et Van Campenhout, 2004 ; Booij et Nguyen, 2005 ; McKeenoch, Liu et Bushnell, 2006). La solution proposée est de s'assurer que tous les neu-

rones du réseau auront la chance d'être souvent actifs. Ceci peut s'effectuer par un choix judicieux des paramètres de départ du réseau. Cependant, il est possible qu'en cours d'apprentissage un neurone devienne silencieux, ce qui est souhaitable si telle est la solution optimale. Mais puisqu'on ne peut pas savoir s'il s'agit bien de la solution optimale, il faut pouvoir continuer l'apprentissage, et donc pouvoir sortir le neurone silencieux de son état. Ainsi, la solution proposée comporte également une procédure d'abaissement graduel du seuil des neurones silencieux durant l'apprentissage.

5.2 Modification à la sortie du quantron

Pour régler le problème des neurones silencieux, nous envisageons de modifier la sortie du quantron de manière à ce que celle-ci ne soit plus discontinue, tout en préservant ses propriétés. Nous proposons la sortie suivante comme exemple de modification :

$$\alpha(x_1, x_2) = t_s(x_1, x_2) + e^{-k(fm(x_1, x_2) - \Gamma)} \quad (5.11)$$

où t_s est une approximation toujours définie de la sortie du quantron, qui sera déterminée, et où k est une constante réelle positive. Cette sortie devrait prendre des valeurs proche de t_s quand le seuil est dépassé, et des valeurs assez grandes quand le seuil n'est pas dépassé.

Observons ce qui se passe avec l'approximation (3.48) lorsqu'une des entrées tend

vers l'infini. Pour l'entrée particulière x , nous avons

$$\begin{bmatrix} a_0(x) \\ a_1(x) \\ a_2(x) \end{bmatrix} = \begin{bmatrix} \frac{2w(4s+x)[\theta^2+(N-1)\theta x+2\theta s]}{x[(N-1)x+2s]^2} \\ \frac{2w(4s+x)[2\theta+(N-1)x+2s]}{x[(N-1)x+2s]^2} \\ \frac{2w(4s+x)}{x[(N-1)x+2s]^2} \end{bmatrix} + \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix}, \quad (5.12)$$

où \vec{v} est la partie des coefficients qui dépend des autres entrées. Lorsque x tend vers l'infini, $\vec{a}(x)$ tend vers \vec{v} :

$$\lim_{x \rightarrow \infty} a_2(x) = v_2 + \lim_{x \rightarrow \infty} \frac{8ws}{x[(N-1)x+2s]^2} + \lim_{x \rightarrow \infty} \frac{2w}{[(N-1)x+2s]^2} = v_2 \quad (5.13)$$

$$\begin{aligned} \lim_{x \rightarrow \infty} a_1(x) &= \\ v_1 + \lim_{x \rightarrow \infty} \frac{16ws(\theta+s)}{x[(N-1)x+2s]^2} + \lim_{x \rightarrow \infty} \frac{8(N-1)ws+4w(\theta+s)}{[(N-1)x+2s]^2} & (5.14) \\ + \lim_{x \rightarrow \infty} \frac{2(N-1)w}{(N-1)^2x+4(N-1)s+4s^2x^{-1}} &= v_1 \end{aligned}$$

$$\begin{aligned} \lim_{x \rightarrow \infty} a_0(x) &= \\ v_0 + \lim_{x \rightarrow \infty} \frac{8ws\theta(\theta+2s)}{x[(N-1)x+2s]^2} + \lim_{x \rightarrow \infty} \frac{8(N-1)ws\theta+2w\theta(\theta+2s)}{[(N-1)x+2s]^2} & (5.15) \\ + \lim_{x \rightarrow \infty} \frac{2(N-1)w\theta}{(N-1)^2x+4(N-1)s+4s^2x^{-1}} &= v_0 \end{aligned}$$

Ainsi, lorsqu'une entrée tend vers l'infini, la fonction d'état du quantron tend à devenir indépendante de cette entrée (et des paramètres qui y sont associés). Ceci correspond à la spécification donnée par les équations (5.4-5.6).

Il reste à définir une approximation appropriée de la sortie du quantron. Nous considérons ici aussi que w_1 et w_2 sont positifs. Nous rappelons que t_m , le temps où le maximum de l'approximation polynomiale est atteint, a été défini en (4.1). Une approximation possible de la sortie du quantron est le temps de croisement entre le seuil Γ et la droite passant par les points (t_m, f_m) et $(t_0, 0)$, avec

$$t_0 = \frac{a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2}, \quad (5.16)$$

qui est la plus petite racine de (3.48). Cette droite, définie par

$$d(t) = \frac{(t - t_0)f_m}{t_m - t_0}, \quad (5.17)$$

croise le seuil au temps

$$t_s = t_0 + \frac{(t_m - t_0)\Gamma}{f_m}. \quad (5.18)$$

Ce temps est toujours défini. Quand $f_m > \Gamma$, t_s est compris entre α , tel que défini en (4.3), et t_m . Quand $f_m < \Gamma$, t_s est plus grand que t_m , ce qui contribue à augmenter la valeur de la sortie, en plus du terme exponentiel.

CHAPITRE 6

ANALYSE ET DISCUSSION

Nous effectuons maintenant un retour sur les différents éléments développés dans ce travail, et par la suite nous en ferons la synthèse.

6.1 Approximations polynomiales

L'approximation polynomiale de la fonction d'état reproduit certains aspects des surfaces discriminantes du quantron. Cependant, nous avons restreint l'étude des propriétés de cette technique d'approximation aux cas où w_1 et w_2 sont positifs. Les approximations se comportent moins bien si on permet à ces paramètres de prendre des valeurs négatives. La présence de poids négatifs peut causer des différences additionnelles entre le modèle mathématique du quantron et son approximation. Il serait donc utile de raffiner l'approximation en utilisant des fonctions plus complexes, comme des fonctions sigmoïdes, qui permettent une approximation beaucoup plus précise des PPS du quantron.. Le calcul analytique du maximum de la fonction d'état ne serait peut-être plus possible, mais il pourrait être évalué numériquement (avec une certaine erreur). L'apprentissage par les fonctions discriminantes ne serait certainement plus possible, mais la descente du gradient pourrait toujours être effectuée, en utilisant l'approximation numérique des dérivées. Il est évident qu'un tel algorithme d'apprentissage serait beaucoup plus lourd en calculs.

6.2 Apprentissage par les fonctions discriminantes

La simplicité de l'approximation des PPS du quantron permet le développement d'une fonction discriminante qui peut être entraînée par l'algorithme d'apprentissage du perceptron, puis par l'inversion d'un système d'équations non linéaire. Ce résultat est intéressant du point de vue de l'étude des fonctions discriminantes en reconnaissance de forme. En effet, une surface conique de la forme (4.9) nécessite habituellement six paramètres. Le modèle, inspiré du quantron, que nous avons développé ne comporte que quatre paramètres (β_1 , β_2 , s et δ), et permet de définir des surfaces discriminantes elliptiques. Le Théorème 1 permet l'apprentissage analytique de ce modèle, mais avec une certaine erreur. En remplaçant ces paramètres par β_1 , r , s et δ , le Théorème 1 permet un apprentissage exact, dans la mesure où r est déterminé auparavant par une autre méthode. Comme il s'agit d'un seul paramètre, la recherche de r peut s'effectuer en une seule dimension, ce qui ne pose pas de difficulté particulière.

6.3 Apprentissage par la descente du gradient

L'apprentissage de l'approximation des signaux du quantron peut être effectué par la descente du gradient. La forme de la fonction d'erreur proposée, inspirée du fonctionnement du quantron, ne peut garantir un apprentissage exact, même pour un seul quantron. Pour cela, il faudrait utiliser l'erreur des moindres carrés avec une fonction discriminante linéaire. Or, nous avons développé la forme linéaire de la fonction discriminante dans le lemme 4, et son application serait difficile à cause du système d'équations mis en cause. L'utilisation d'une forme plus complexe de fonction d'erreur permet tout de même d'obtenir des résultats intéressants. Ainsi, l'adaptation

de l'algorithme de la rétro-propagation de l'erreur aux réseaux de quantrons semble réalisable. Cette adaptation permettra ensuite d'appliquer diverses techniques qui améliorent la vitesse de convergence de la rétro-propagation, et l'intégration d'heuristiques de recherche de minimum global qui sont souvent utilisées dans les réseaux de neurones classiques.

6.4 Réseaux de quantrons

La combinaison de plusieurs quantrons en réseau introduit de nouvelles difficultés dans le développement d'un algorithme d'apprentissage. Nous avons proposé une modification à la sortie du quantron qui rend sa fonction d'activation continûment dérivable. D'autres modifications pourraient être envisagées. Par exemple, nous pourrions développer une modification par l'utilisation directe du discriminant $a_1^2 - 4a_2(a_0 + \Gamma)$ qui indique si le seuil est dépassé ou non. L'étude de la performance de ces modifications, en comparaison à l'approximation avec une sortie discontinue, et avec le modèle original du quantron, reste cependant à faire.

6.5 Discussion

Nous avons pu vérifier que la simplification du modèle mathématique du quantron permet d'établir des méthodes d'apprentissage pour un seul neurone. Puisque l'apprentissage exact nécessite une grande réduction de la complexité du quantron, il faut tolérer une certaine erreur lors de l'entraînement pour profiter d'un modèle plus puissant. La recherche d'une approximation et d'une fonction d'erreur permettant un apprentissage avec une erreur faible est donc primordiale pour développer un algorithme d'apprentissage pour le quantron.

Les réseaux de quantrons ont une puissance beaucoup plus grande que celle d'un simple quantron. Cependant, l'adaptation de l'algorithme de rétro-propagation demande de modifier la sortie du quantron (tout comme les fonctions sigmoïdes qui remplacent les fonctions Heaviside dans les réseaux de perceptrons). Cette modification peut causer des divergences avec le modèle original. Les recherches futures devront déterminer comment minimiser l'impact de cette modification.

CHAPITRE 7

CONCLUSION

Cette recherche nous a permis d'explorer les caractéristiques de deux approximations polynomiales de la fonction d'état du quantron. Nous avons identifié le potentiel et les limites de ces approximations en utilisant deux approches différentes à l'apprentissage. La première approximation, qui porte sur les potentiels post-synaptiques du quantron, permet un apprentissage par l'étude directe de sa fonction discriminante. La deuxième approximation, qui porte sur les signaux d'entrée du quantron, peut être entraînée par la descente du gradient sur une fonction d'erreur appropriée.

La deuxième approximation s'est avérée plus efficace que la première pour reproduire les surfaces discriminantes du quantron. Pour adapter l'algorithme de rétro-propagation de l'erreur à un réseau de quantrons, nous avons démontré la nécessité de modifier la sortie du quantron pour que sa fonction d'activation soit continûment dérivable.

Nos travaux contribuent à fournir les éléments de base nécessaires au développement d'un algorithme d'apprentissage rapide pour le quantron. De plus, l'étude de la première méthode d'approximation a démontré la possibilité d'effectuer l'apprentissage d'une classe restreinte de formes elliptiques avec quatre paramètres (au lieu de six pour les coniques en général).

Diverses questions soulevées par cette recherche constituent la suite de nos travaux. Premièrement, la recherche d'approximations plus complexes pour le quantron permettrait de mieux reproduire les surfaces discriminantes du quantron. Deuxièmement, les théorèmes développés sur la première approximation devraient être abordés d'un point de vue géométrique pour tenter de trouver une méthode efficace pour déterminer le paramètre r . Finalement, l'adaptation de l'algorithme de rétro-propagation de l'erreur au quantron nécessite l'étude en profondeur des fonctions d'erreur et des modifications de la sortie du quantron.

RÉFÉRENCES

BARTLETT, P.L., et DOWNS, T. (1992). Using random weights to train multilayer networks of hard-limiting units. *IEEE Transactions on Neural Networks*, 3(2), 202-210.

BATRUNI, R. (1991). A multilayer neural network with piecewise-linear structure and back-propagation learning. *IEEE Transactions on Neural Networks*, 2(3), 395-403.

BOOIJ, O., et NGUYEN, H.T. (2005). A gradient descent rule for spiking neurons emitting multiple spikes. *Information Processing Letters*, 95, 552-558.

BOTHE, S.M., KOK, J.N., et LA POUTRÉ, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48, 17-37.

BURGSTEINER, H. (2005). Training networks of biological realistic spiking neurons for real-time robot control. *Proceedings of the 9th International Conference on Engineering Applications of Neural Networks*, 129-136.

CARNELL, A., et RICHARDSON, D. (2005). Linear algebra for time series of spikes. *ESANN'2005 proceedings*, 363-368.

DAVIS, B.A., ERDOGMUS, D., RAO, Y.N., et PRINCIPE, J.C. (2003). Supervised synaptic weight adaptation for a spiking neuron. *Proceedings of the International Joint Conference on Neural Networks*, 4, 2558-2562.

DURBIN, R., et RUMELHART, D. (1989). Product units : A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation*, 1, 133-142.

GERSTNER, W. (1995). Time structure activity in neural network models. *Physical Review E*, 51(1), 738-758.

GERSTNER, W., KEMPTER, R., VAN HEMMEN, J.L., et WAGNER, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 384, 76-78.

GERSTNER, W., et KISTLER, W.M. (2002). Mathematical formulations of Hebbian learning. *Biological Cybernetics*, 87, 404-415.

GOODMAN, R.M., et ZENG, Z. (1994). A learning algorithm for multi-layer perceptrons with hard-limiting threshold units. *Proceedings of the 1994 IEEE Workshop on Neural Networks for Signal Processing*, 219-228.

HAYKIN, S. (1999). *Neural networks : a comprehensive foundation* (2^e éd.), Upper Saddle River, NJ : Prentice Hall.

LABIB, R. (1999). New single neuron structure for solving nonlinear problems. *Proceedings of the International Joint Conference on Neural Networks*, 1, 617-620.

LABIB, R., et ASSADI, R. (2007). Modified multi-layered perceptron applied to packing and covering problems. *Neural Computing and Applications*, 162, 173-186.

LECUN, Y., BOSER, B., DENKER, J.S., HENDERSON, D., HOWARD, R.E., HUBBARD, W. et JACKEL, L.D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing*, 2, 396-404, San Mateo, CA : Morgan Kaufmann.

MAASS, W. (1997). Networks of spiking neurons : the third generation of neural network models. *Neural Networks*, 10(9), 1659-1671.

MAASS, W., NATSCHLÄGER, T., et MARKRAM, H. (2002). Real-time computing without stable states : a new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531-2560.

MAASS, W., et SCHMITT, M. (1999). On the complexity of learning for spiking neurons with temporal coding. *Information and Computation*, 153, 26-46.

MCKENNOCH, S., LIU, D., et BUSHNELL, L.G. (2006) Fast Modifications of the SpikeProp Algorithm. *IEEE World Congress on Computational Intelligence*.

MEGIDDO, N. (1984). Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1), 114-127.

MIYAMOTO, H., KAWATO, M., SETOYAMA, T., et SUZUKI, R. (1988). Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator. *Neural Networks*, 1, 251-265.

NOCEDAL, J., et WRIGHT, S.J. (1999). *Numerical Optimization*, New York : Springer.

NILSSON, N.J. (1965). *Learning Machines : Foundations of Trainable Pattern Classifying Systems*, New York : McGraw-Hill.

PAVLIDIS, N.G., TASOULIS, D.K., PLAGIANAKOS, V.P., NIKIFORIDIS, G., et VRAHITIS, M.N. (2005). Spiking neural network training using evolutionary algorithms. *Proceedings of the International Joint Conference on Neural Networks*, 4, 2190-2194.

REDDING, N.J., et DOWNS, T. (1991). Learning in feedforward networks with nonsmooth functions : an ℓ_∞ example. *1991 IEEE International Joint Conference on Neural Networks*, 2, 947-952.

RIDELLA, S., ROVETTA, S. et ZUNINO, R. (1997) Circular Backpropagation Networks for Classification. *IEEE Transactions on Neural Networks*, 8(1), 84-97.

RUMELHART, J.L., HINTON, G.E., et WILLIAMS, R.J. (1986). Learning representations by back-propagation of errors. *Nature*, 323, 533-536.

RUF, B., et SCHMITT, M. (1997). Learning temporally encoded patterns in networks of spiking neurons. *Neural Processing Letters*, 5(1), 9-18.

SCHRAUWEN, B. , et VAN CAMPENHOUT, J. (2004). Extending Spikeprop. *Proceedings of the International Joint Conference on Neural Networks*, 1, 471-475.

ZHANG, Z., et SARHADI, M. (1993). A modified neuron activation function which enables single layer perceptrons to solve some linearly inseparable problems. *Proceedings of the International Joint Conference on Neural Networks*, 3, 2723-2726.

ANNEXE A

LEMME 4 : COEFFICIENTS DÉTAILLÉS

Les coefficients détaillés de la fonction discriminante (4.50) ont été évalués avec le logiciel **Maple**.

$$e_{55} = (N - 1)^6 w_1 w_2 \quad (\text{A.1})$$

$$e_{54} = 4(N - 1)^4 s_1 w_2^2 (4s_1 + N - 1) - 64\Gamma(N - 1)^5 s_1 w_2 + 2(N - 1)^5 w_1 w_2 [(2N + 1)s_2 + \theta_2 - 2\theta_1] \quad (\text{A.2})$$

$$e_{53} = 16(N - 1)^3 s_1 w_2^2 [(2N - 1)s_2 - 3\theta_2] (4s_1 + N - 1) + 4(N - 1)^4 s_2 w_1 w_2 [3(2N - 1)s_2 + 2N(\theta_2 - 2\theta_1)] - 256\Gamma N (N - 1)^4 s_1 s_2 w_2 \quad (\text{A.3})$$

$$e_{52} = 16(N - 1)^3 s_1 w_2^2 [(4N^2 - 3)s_2^2 - 2s_2\theta_2 - \theta_2^2] + 64(N - 1)^2 s_1^2 w_2^2 [(4N^2 - 3)s_2^2 - 2s_2\theta_2 - \theta_2^2] + 8(N - 1)^3 s_2^2 w_1 w_2 [(6N - 5)s_2 + (4N - 3)\theta_2 - 2(4N - 3)\theta_1] - 256\Gamma(4N - 3)(N - 1)^3 s_1 s_2^2 w_2 \quad (\text{A.4})$$

$$e_{51} = 128(N - 1)^2 w_2^2 s_1 s_2^2 [(2N - 1)s_2 - 2(N - 1)\theta_2] (4s_1 + N - 1) + 32(N - 1)^3 w_1 w_2 s_2^3 (s_2 + \theta_2 - 2\theta_1) - 1024\Gamma(N - 1)^3 w_2 s_1 s_2^3 \quad (\text{A.5})$$

$$\begin{aligned}
e_{45} = & 8(N-1)^5 s_2 w_1^2 + 2(N-1)^5 w_1 w_2 [(2N-1)s_1 + \theta_1 - 2\theta_2] \\
& + 4(N-1)^5 s_1 w_1 w_2 - 64\Gamma(N-1)^5 w_1 s_2
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
e_{44} = & 24(N-1)^4 s_2^2 w_1^2 + 24(N-1)^4 s_1^2 w_2^2 + 16(N-1)^4 s_1 s_2 w_1 w_2 \\
& + 8(N-1)^4 w_1 w_2 [2(2N-1)s_1 s_2 + s_1 \theta_2 + s_2 \theta_1 - 2(s_1 \theta_1 + s_2 \theta_2)] \\
& + 4(2N-1)(N-1)^4 w_1 w_2 (s_1 \theta_2 + s_2 \theta_1 - 2s_1 \theta_1 - 2s_2 \theta_2) \\
& + 4(N-1)^4 w_1 w_2 [(2N-1)^2 s_1 s_2 + \theta_1 \theta_2 - \theta_1^2 - \theta_2^2] \\
& - 192\Gamma(N-1)^4 s_2^2 w_1 - 192\Gamma(N-1)^4 s_1^2 w_2
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
e_{43} = & 96(N-1)^3 w_2^2 s_1^2 [(2N-1)s_2 - 3\theta_2] + 16(N-1)^3 w_1 w_2 s_1 s_2^2 \\
& + 32(N-1)^3 w_1 w_2 s_1 s_2 [(3N-2)s_2 + N\theta_2 - 2N\theta_1] \\
& + 8(N-1)^3 s_2^2 w_1 w_2 [(2N-1)s_1 + \theta_1 - 2\theta_2] \\
& + 16(2N-1)(N-1)^3 w_1 w_2 s_2 (s_1 \theta_2 + s_2 \theta_1 - 2s_1 \theta_1 - 2s_2 \theta_2) \\
& + 16(N-1)^3 w_1 w_2 s_2 [(2N-1)^2 s_1 s_2 + \theta_1 \theta_2 - \theta_1^2 - \theta_2^2] \\
& + 16(2N-1)(N-1)^4 w_1 w_2 s_1 s_2 (s_2 + \theta_2 - 2\theta_1) \\
& + 16(N-1)^4 w_1 w_2 [s_2^2 (\theta_1 - 4\theta_2) - s_2 (\theta_1^2 + 2\theta_2^2) + s_1 s_2 \theta_1 \theta_2] \\
& + 32(N-1)^3 w_1^2 s_2^3 - 768\Gamma N(N-1)^3 w_2 s_1^2 s_2 - 256\Gamma(N-1)^3 w_1 s_2^3
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
e_{42} = & 16(N-1)^2 w_1^2 s_2^4 + 96(N-1)^2 w_2^2 s_1^2 ((4N^2 - 3)s_2^2 - 2s_2 \theta_2 - \theta_2^2) \\
& + 32(N-1)^2 w_1 w_2 s_1 s_2^2 [(6N-5)s_2 + (4N-3)\theta_2 - 2(4N-3)\theta_1] \\
& + 16(2N-1)(N-1)^2 w_1 w_2 s_2^2 (s_1 \theta_2 + s_2 \theta_1 - 2s_1 \theta_1 - 2s_2 \theta_2) \\
& + 16(N-1)^2 w_1 w_2 s_2^2 [(2N-1)^2 s_1 s_2 + \theta_1 \theta_2 - \theta_1^2 - \theta_2^2] \\
& + 64(2N-1)(N-1)^3 w_1 w_2 s_1 s_2^2 (s_2 + \theta_2 - 2\theta_1) \\
& + 64(N-1)^3 w_1 w_2 s_2 [s_2^2 (\theta_1 - 4\theta_2) - s_2 (\theta_1^2 + 2\theta_2^2) + s_1 s_2 \theta_1 \theta_2] \\
& - 768\Gamma(4N-3)(N-1)^2 w_2 s_1^2 s_2^2 - 128\Gamma(N-1)^2 w_1 s_2^4
\end{aligned} \tag{A.9}$$

$$\begin{aligned}
e_{41} = & 768(N-1)^2 w_2^2 s_1^2 s_2^2 ((2N-1)s_2 - 2(N-1)\theta_2) \\
& + 64(2N+1)(N-1)^2 w_1 w_2 s_1 s_2^3 (s_2 + \theta_2 - 2\theta_1) \\
& + 64(N-1)^2 w_1 w_2 s_2^2 [s_2^2 (\theta_1 - 4\theta_2) - s_2 (\theta_1^2 + 2\theta_2^2) + s_1 s_2 \theta_1 \theta_2] \\
& - 3072\Gamma(N-1)^2 w_2 s_1^2 s_2^3
\end{aligned} \tag{A.10}$$

$$\begin{aligned}
e_{35} = & 32(N-1)^4 w_1^2 s_2 [(2N-1)s_1 - 3\theta_1] + 4(N-1)^4 w_1 w_2 s_1^2 \\
& + 8(N-1)^4 w_1 w_2 s_1 [(3N-2)s_1 + N\theta_1 - 2N\theta_2] \\
& - 256\Gamma N(N-1)^4 w_1 s_1 s_2
\end{aligned} \tag{A.11}$$

$$\begin{aligned}
e_{34} = & 32(N-1)^3 w_2^2 s_1^3 + 96(N-1)^3 w_1^2 s_2^2 [(2N-1)s_1 - 3\theta_1] \\
& + 16(N-1)^3 w_1 w_2 s_1^2 s_2 + 8(N-1)^3 w_1 w_2 s_1^2 [(2N-1)s_2 + \theta_2 - 2\theta_1] \\
& + 32(N-1)^3 w_1 w_2 s_1 s_2 [(3N-2)s_1 + N\theta_1 - 2N\theta_2] \\
& + 16(2N-1)(N-1)^3 s_1 w_1 w_2 (s_1 \theta_2 + s_2 \theta_1 - 2s_1 \theta_1 - 2s_2 \theta_2) \\
& + 16(N-1)^3 s_1 w_1 w_2 [(2N-1)^2 s_1 s_2 + \theta_1 \theta_2 - \theta_1^2 - \theta_2^2] \\
& + 16(2N-1)(N-1)^4 w_1 w_2 s_1 s_2 (s_1 + \theta_1 - 2\theta_2) \\
& + 16(N-1)^4 w_1 w_2 [s_1^2 (\theta_2 - 4\theta_1) - s_1 (\theta_2^2 + 2\theta_1^2) + s_1 s_2 \theta_1 \theta_2] \\
& - 768\Gamma N(N-1)^3 w_1 s_1 s_2^2 - 256\Gamma(N-1)^3 w_2 s_1^3
\end{aligned} \tag{A.12}$$

$$\begin{aligned}
e_{33} = & 128(N-1)^2 w_1^2 s_2^3 [(2N-1)s_1 - 3\theta_1] + 128(N-1)^2 w_2^2 s_1^3 [(2N-1)s_2 - 3\theta_2] \\
& + 32(N-1)^2 w_1 w_2 s_1 s_2 [2(2N-1)s_1 s_2 + s_1(\theta_2 - 2\theta_1) + s_2(\theta_1 - 2\theta_2)] \\
& + 32(4N-1)(N-1)^3 w_1 w_2 s_1 s_2 [s_1(s_2 + \theta_2 - 2\theta_1) + s_2(s_1 + \theta_1 - 2\theta_2)] \\
& + 64(2N-1)(N-1)^2 w_1 w_2 s_1 s_2 (s_1 \theta_2 + s_2 \theta_1 - 2s_1 \theta_1 - 2s_2 \theta_2) \\
& + 64(N-1)^2 w_1 w_2 s_1 s_2 [(2N-1)^2 s_1 s_2 + \theta_1 \theta_2 - \theta_1^2 - \theta_2^2] \\
& + 64(N-1)^3 w_1 w_2 s_1 s_2 [s_2(\theta_1 - 4\theta_2) - \theta_1^2 - 2\theta_2^2 + s_1 \theta_1 \theta_2] \\
& + 64(N-1)^3 w_1 w_2 s_1 s_2 [s_1(\theta_2 - 4\theta_1) - \theta_2^2 - 2\theta_1^2 + s_2 \theta_1 \theta_2] \\
& + 64(N-1)^4 w_1 w_2 s_1 s_2 (s_1 s_2 + \theta_1 \theta_2 + s_1 \theta_2 + s_2 \theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1 \theta_1 - 4s_2 \theta_2) \\
& + 16(N-1)^2 w_1 w_2 s_1^2 s_2^2 - 1024\Gamma N(N-1)^2 w_1 s_1 s_2^3 - 1024\Gamma N(N-1)^2 w_2 s_1^3 s_2
\end{aligned} \tag{A.13}$$

$$\begin{aligned}
e_{32} = & 64(N-1)w_1^2 s_2^4 [(2N-1)s_1 - 3\theta_1] \\
& + 128(N-1)w_2^2 s_1^3 [(4N^2 - 3)s_2^2 - 2s_2 \theta_2 - \theta_2^2] \\
& + 32(N-1)w_1 w_2 s_1^2 s_2^2 [(6N-5)s_2 + (4N-3)\theta_2 - 2(4N-3)\theta_1] \\
& + 64(2N-1)(N-1)w_1 w_2 s_1 s_2^2 (s_1 \theta_2 + s_2 \theta_1 - 2s_1 \theta_1 - 2s_2 \theta_2) \\
& + 64(N-1)w_1 w_2 s_1 s_2^2 [(2N-1)^2 s_1 s_2 + \theta_1 \theta_2 - \theta_1^2 - \theta_2^2] \\
& + 256(2N-1)(N-1)^2 w_1 w_2 s_1^2 s_2^2 (s_2 + \theta_2 - 2\theta_1) \\
& + 256(N-1)^2 w_1 w_2 s_1 s_2 [s_2^2(\theta_1 - 4\theta_2) - s_2(\theta_1^2 + 2\theta_2^2) + s_1 s_2 \theta_1 \theta_2] \\
& + 64(2N-1)(N-1)^2 w_1 w_2 s_1 s_2^3 (s_1 + \theta_1 - 2\theta_2) \\
& + 64(N-1)^2 w_1 w_2 s_2^2 [s_1^2(\theta_2 - 4\theta_1) - s_1(\theta_2^2 + 2\theta_1^2) + s_1 s_2 \theta_1 \theta_2] \\
& + 256(N-1)^3 w_1 w_2 s_1 s_2^2 (s_1 s_2 + \theta_1 \theta_2 + s_1 \theta_2 + s_2 \theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1 \theta_1 - 4s_2 \theta_2) \\
& - 1024\Gamma(4N-3)(N-1)w_2 s_1^3 s_2^2 - 512\Gamma N(N-1)w_1 s_1 s_2^4
\end{aligned} \tag{A.14}$$

$$\begin{aligned}
e_{31} = & 1024(N-1)w_2^2s_1^3s_2^2[(2N-1)s_2 - 2(N-1)\theta_2] \\
& +128(N-1)w_1w_2s_1^2s_2^3(s_2 + \theta_2 - 2\theta_1) \\
& +256(2N-1)(N-1)w_1w_2s_1^2s_2^3(s_2 + \theta_2 - 2\theta_1) \\
& +256(N-1)s_1s_2^2w_1w_2[s_2^2(\theta_1 - 4\theta_2) - s_2(\theta_1^2 + 2\theta_2^2) + s_1s_2\theta_1\theta_2] \quad (\text{A.15}) \\
& +256(N-1)^2w_1w_2s_1s_2^3(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 \\
& -2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \\
& -4096\Gamma(N-1)w_2s_1^3s_2^3
\end{aligned}$$

$$\begin{aligned}
e_{25} = & 32(N-1)^3w_1^2s_2[(4N^2-3)s_1^2 - 2s_1\theta_1 - \theta_1^2] \\
& +8(N-1)^3w_1w_2s_1^2[(6N-5)s_1 + (4N-3)\theta_1 - 2(4N-3)\theta_2] \quad (\text{A.16}) \\
& -256\Gamma(4N-3)(N-1)^3w_1s_1^2s_2
\end{aligned}$$

$$\begin{aligned}
e_{24} = & 16(N-1)^2w_2^2s_1^4 + 96(N-1)^2w_1^2s_2^2[(4N^2-3)s_1^2 - 2s_1\theta_1 - \theta_1^2] \\
& +32(N-1)^2w_1w_2s_1^2s_2[(2N-1)s_1 + \theta_1 - 2\theta_2] \\
& +16(2N-1)(N-1)^2w_1w_2s_1^2(s_1\theta_2 + s_2\theta_1 - 2s_1\theta_1 - 2s_2\theta_2) \\
& +16(N-1)^2w_1w_2s_1^2[(2N-1)^2s_1s_2 + \theta_1\theta_2 - \theta_1^2 - \theta_2^2] \quad (\text{A.17}) \\
& +64(2N+1)(N-1)^3w_1w_2s_1^2s_2(s_1 + \theta_1 - 2\theta_2) \\
& +64(N-1)^3w_1w_2s_1(s_1^2(\theta_2 - 4\theta_1) - s_1(\theta_2^2 + 2\theta_1^2) + s_1s_2\theta_1\theta_2) \\
& -768\Gamma(4N-3)(N-1)^2w_1s_1^2s_2^2 - 128\Gamma(N-1)^2w_2s_1^4
\end{aligned}$$

$$\begin{aligned}
e_{23} = & 64(N-1)w_2^2s_1^4[(2N-1)s_2 - 3\theta_2] \\
& +128(N-1)w_1^2s_2^3[(4N^2-3)s_1^2 - 2s_1\theta_1 - \theta_1^2] \\
& +32(N-1)w_1w_2s_1^2s_2^2[(2N-1)s_1 + \theta_1 - 2\theta_2] \\
& +64(2N-1)(N-1)w_1w_2s_1^2s_2(s_1\theta_2 + s_2\theta_1 - 2s_1\theta_1 - 2s_2\theta_2) \\
& +64(N-1)w_1w_2s_1^2s_2[(2N-1)^2s_1s_2 + \theta_1\theta_2 - \theta_1^2 - \theta_2^2] \\
& +64(2N-1)(N-1)^2w_1w_2s_1^3s_2(s_2 + \theta_2 - 2\theta_1) \\
& +64(N-1)^2w_1w_2s_1^2s_2[s_2(\theta_1 - 4\theta_2) - \theta_1^2 - 2\theta_2^2 + s_1\theta_1\theta_2] \\
& +128(4N-1)(N-1)^2w_1w_2s_1^2s_2^2(s_1 + \theta_1 - 2\theta_2) \\
& +256(N-1)^2w_1w_2s_1s_2[s_1^2(\theta_2 - 4\theta_1) - s_1(\theta_2^2 + 2\theta_1^2) + s_1s_2\theta_1\theta_2] \\
& +256(N-1)^3w_1w_2s_1^2s_2(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \\
& -1024(4N-3)\Gamma(N-1)w_1s_1^2s_2^3 - 512\Gamma N(N-1)w_2s_1^4s_2
\end{aligned} \tag{A.18}$$

$$\begin{aligned}
e_{22} = & 64w_2^2s_1^4[(4N^2-3)s_2^2 - 2s_2\theta_2 - \theta_2^2] + 64w_1^2s_2^4[(4N^2-3)s_1^2 - 2s_1\theta_1 - \theta_1^2] \\
& +64(2N-1)w_1w_2s_1^2s_2^2(s_1\theta_2 + s_2\theta_1 - 2s_1\theta_1 - 2s_2\theta_2) \\
& +64w_1w_2s_1^2s_2^2[(2N-1)^2s_1s_2 + \theta_1\theta_2 - \theta_1^2 - \theta_2^2] \\
& +256(2N-1)(N-1)w_1w_2s_1^3s_2^2(s_2 + \theta_2 - 2\theta_1) \\
& +256(N-1)w_1w_2s_1^2s_2^2[s_2(\theta_1 - 4\theta_2) - \theta_1^2 - 2\theta_2^2 + s_1\theta_1\theta_2] \\
& +256(2N-1)(N-1)w_1w_2s_1^2s_2^3(s_1 + \theta_1 - 2\theta_2) \\
& +256(N-1)w_1w_2s_1^2s_2^2[s_1(\theta_2 - 4\theta_1) - \theta_2^2 - 2\theta_1^2 + s_2\theta_1\theta_2] \\
& +1024(N-1)^2w_1w_2s_1^2s_2^2(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \\
& -512\Gamma(4N-3)w_2s_1^4s_2^2 - 512\Gamma(4N-3)w_1s_1^2s_2^4
\end{aligned} \tag{A.19}$$

$$\begin{aligned}
e_{21} = & 512w_2^2s_1^4s_2^2[(2N-1)s_2 - 2(N-1)\theta_2] \\
& + 256(2N-1)w_1w_2s_1^3s_2^3(s_2 + \theta_2 - 2\theta_1) \\
& + 256w_1w_2s_1^2s_2^3[s_2(\theta_1 - 4\theta_2) - \theta_1^2 - 2\theta_2^2 + s_1\theta_1\theta_2] \\
& + 1024(N-1)w_1w_2s_1^2s_2^3(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \\
& - 2048\Gamma w_2s_1^4s_2^3
\end{aligned} \tag{A.20}$$

$$\begin{aligned}
e_{15} = & 256(N-1)^3w_1^2s_1^2s_2[(2N-1)s_1 - 2(N-1)\theta_1] \\
& + 32(N-1)^3w_1w_2s_1^3(s_1 + \theta_1 - 2\theta_2) - 1024\Gamma(N-1)^3w_1s_1^3s_2
\end{aligned} \tag{A.21}$$

$$\begin{aligned}
e_{14} = & 768(N-1)^2w_1^2s_1^2s_2^2[(2N-1)s_1 - 2(N-1)\theta_1] \\
& + 64(2N+1)(N-1)^2w_1w_2s_1^3s_2(s_1 + \theta_1 - 2\theta_2) \\
& + 64(N-1)^2w_1w_2s_1^3(s_1(\theta_2 - 4\theta_1) - \theta_2^2 - 2\theta_1^2 + s_2\theta_1\theta_2) \\
& - 3072\Gamma(N-1)^2w_1s_1^3s_2^2
\end{aligned} \tag{A.22}$$

$$\begin{aligned}
e_{13} = & 1024(N-1)w_1^2s_1^2s_2^3[(2N-1)s_1 - 2(N-1)\theta_1] \\
& + 128(4N-1)w_1w_2s_1^3s_2^2(s_1 + \theta_1 - 2\theta_2) \\
& + 256(N-1)w_1w_2s_1^3s_2[s_1(\theta_2 - 4\theta_1) - \theta_2^2 - 2\theta_1^2 + s_2\theta_1\theta_2] \\
& + 256(N-1)^2w_1w_2s_1^3s_2(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \\
& - 4096\Gamma(N-1)w_1s_1^3s_2^3
\end{aligned} \tag{A.23}$$

$$\begin{aligned}
e_{12} = & 512s_2^4w_1^2s_1^2[(2N-1)s_1 - 2(N-1)\theta_1] \\
& + 256(2N-1)w_1w_2s_1^3s_2^3(s_1 + \theta_1 - 2\theta_2) \\
& + 256w_1w_2s_1^3s_2^2[s_1(\theta_2 - 4\theta_1) - \theta_2^2 - 2\theta_1^2 + s_2\theta_1\theta_2] \\
& + 1024(N-1)w_1w_2s_1^3s_2^2(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 \\
& - 2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \\
& - 2048\Gamma w_1s_1^3s_2^4
\end{aligned} \tag{A.24}$$

$$e_{11} = 1024s_1^3s_2^3w_1w_2(s_1s_2 + \theta_1\theta_2 + s_1\theta_2 + s_2\theta_1 - 2\theta_1^2 - 2\theta_2^2 - 4s_1\theta_1 - 4s_2\theta_2) \tag{A.25}$$

$$e_{60} = 64(N-1)^4w_2^2s_2^2(s_2^2 - 6s_2\theta_2 - 3\theta_2^2) \tag{A.26}$$

$$e_{50} = 512(N-1)^3w_2^2s_1s_2^2(s_2^2 - 6s_2\theta_2 - 3\theta_2^2) \tag{A.27}$$

$$e_{40} = 1536(N-1)^2s_1^2w_2^2s_2^2(s_2^2 - 6s_2\theta_2 - 3\theta_2^2) \tag{A.28}$$

$$e_{30} = 2048s_1^3(N-1)w_2^2s_2^2(s_2^2 - 6s_2\theta_2 - 3\theta_2^2) \tag{A.29}$$

$$e_{20} = 1024s_1^4w_2^2s_2^2(s_2^2 - 6s_2\theta_2 - 3\theta_2^2) \tag{A.30}$$

$$e_{02} = 1024s_2^4w_1^2s_1^2(s_1^2 - 6s_1\theta_1 - 3\theta_1^2) \tag{A.31}$$

$$e_{03} = 2048s_2^3(N-1)w_1^2s_1^2(s_1^2 - 6s_1\theta_1 - 3\theta_1^2) \tag{A.32}$$

$$e_{04} = 1536s_2^2(N-1)^2w_1^2s_1^2(s_1^2 - 6s_1\theta_1 - 3\theta_1^2) \quad (\text{A.33})$$

$$e_{05} = 512(N-1)^3s_2w_1^2s_1^2(s_1^2 - 6s_1\theta_1 - 3\theta_1^2) \quad (\text{A.34})$$

$$e_{06} = 64(N-1)^4w_1^2s_1^2(s_1^2 - 6s_1\theta_1 - 3\theta_1^2) \quad (\text{A.35})$$

$$e_{61} = 32(N-1)^4w_2^2s_2^2((2N-1)s_2 - 2(N-1)\theta_2) - 128\Gamma(N-1)^4s_2^3w_2 \quad (\text{A.36})$$

$$e_{62} = 4(N-1)^4w_2^2[(4N^2 - 3)s_2^2 - 2s_2\theta_2 - \theta_2^2] - 32\Gamma(N-1)^4s_2^2w_2 - 128\Gamma(N-1)^5s_2^2w_2 \quad (\text{A.37})$$

$$e_{63} = 4(N-1)^5w_2^2((2N-1)s_2 - 3\theta_2) - 32\Gamma(N-1)^5s_2w_2 - 32\Gamma(N-1)^6w_2s_2 \quad (\text{A.38})$$

$$e_{64} = (N-1)^6w_2^2 - 8\Gamma(N-1)^6w_2 \quad (\text{A.39})$$

$$e_{16} = 32(N-1)^4w_1^2s_1^2((2N-1)s_1 - 2(N-1)\theta_1) - 128\Gamma s_1^3(N-1)^4w_1 \quad (\text{A.40})$$

$$e_{26} = 4(N-1)^4w_1^2[(4N^2 - 3)s_1^2 - 2s_1\theta_1 - \theta_1^2] - 32\Gamma s_1^2(N-1)^4w_1 - 128\Gamma(N-1)^5s_1^2w_1 \quad (\text{A.41})$$

$$e_{36} = 4(N-1)^5w_1^2((2N-1)s_1 - 3\theta_1) - 32\Gamma(N-1)^5s_1w_1 - 32\Gamma(N-1)^6w_1s_1 \quad (\text{A.42})$$

$$e_{46} = (N-1)^6w_1^2 - 8\Gamma(N-1)^6w_1 \quad (\text{A.43})$$

ANNEXE B

EXPÉRIENCE 1A :

IMPLÉMENTATION

Voici l'implémentation de l'algorithme d'apprentissage du perceptron en **Matlab**. La variable T , chargée avant l'application de l'algorithme, contient trois informations dans chaque ligne : la coordonnée x_1 , la coordonnée x_2 , et la classe désirée.

```
%Chargement du problème
i = 1;
Tm = [];
while i <= 200
    x1 = T(i,1);
    x2 = T(i,2);
    Tm = [Tm;
          (1-N^2)*x1^2/3, (1-N^2)*x2^2/3,
          2*(N-1)^2*x1*x2-(2/3)*(N-1)*(2*N-1)*(x1^2+x2^2),
          4*(N-1)*(x1-x2), T(i,3)];
    i=i+1;
end

err = 1;
c = [0 0 0 0 0];
```

```
iter = 1;

%L'algorithme d'apprentissage du perceptron converge
%vers une erreur nulle
while err > 0
    mod = 0;

    %Choix au hasard d'un exemple
    %jusqu'à ce qu'un exemple mal classé soit trouvé
    while(mod == 0)
        i = randint(1,1,[1 length(Tm)]);
        f = [1,Tm(i,1:4)];
        g = f*c';
        if(g < 0 && Tm(i,5) >= 0.5)
            c = c + 0.01*f;
            mod = 1;
        elseif(g >= 0 && Tm(i,5) < 0.5)
            c = c - 0.01*f;
            mod = 1;
        end
    end
end

%Calcul du nombre d'exemples mal classés
err = 0;
```

```
for i = 1:length(Tm)
    f = [1,Tm(i,1:4)];
    g = f*c';
    if(g < 0 && Tm(i,5) >= 0.5)
        err = err + 1;
    elseif(g >= 0 && Tm(i,5) < 0.5)
        err = err + 1;
    end
end
iter = iter + 1;
if(err < 20)
    iter
    err
end
end
```

À la fin de l'algorithme, le vecteur c contient les poids qui résolvent le problème d'apprentissage.

ANNEXE C

EXPÉRIENCE 2 : RÉSULTATS DÉTAILLÉS

Tableau C.1 – Résultats détaillés de l'expérience 2, $c = 1$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 180.7296 | 47 | 42 | 6 | 148.4532 | 31 | 7 |
| 2 | 141.4536 | 25 | 5 | 7 | 199.4746 | 51 | 48 |
| 3 | 217.1269 | 49 | 47 | 8 | 165.2911 | 33 | 31 |
| 4 | 165.5235 | 33 | 31 | 9 | 146.2389 | 28 | 16 |
| 5 | 141.5597 | 22 | 4 | 10 | 144.1544 | 19 | 16 |

Tableau C.2 – Résultats détaillés de l'expérience 2, $c = 2$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 84.0683 | 9 | 5 | 6 | 149.7496 | 49 | 48 |
| 2 | 106.0729 | 12 | 11 | 7 | 96.4060 | 11 | 5 |
| 3 | 112.2956 | 21 | 16 | 8 | 79.8110 | 12 | 12 |
| 4 | 99.0418 | 10 | 5 | 9 | 93.9135 | 15 | 15 |
| 5 | 84.0917 | 8 | 5 | 10 | 83.7408 | 7 | 6 |

Tableau C.3 – Résultats détaillés de l'expérience 2, $c = 4$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 51.4719 | 6 | 4 | 6 | 41.3106 | 5 | 4 |
| 2 | 56.8414 | 15 | 13 | 7 | 66.1269 | 19 | 19 |
| 3 | 42.4594 | 11 | 7 | 8 | 45.1297 | 5 | 2 |
| 4 | 95.8979 | 30 | 27 | 9 | 54.7165 | 15 | 13 |
| 5 | 55.0432 | 15 | 14 | 10 | 44.6543 | 7 | 7 |

Tableau C.4 – Résultats détaillés de l'expérience 2, $c = 6$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 45.2852 | 14 | 11 | 6 | 137.0511 | 35 | 33 |
| 2 | 53.0397 | 16 | 12 | 7 | 31.3004 | 5 | 5 |
| 3 | 28.7764 | 5 | 4 | 8 | 30.6351 | 5 | 4 |
| 4 | 30.4925 | 5 | 4 | 9 | 38.9111 | 7 | 7 |
| 5 | 29.0115 | 2 | 1 | 10 | 30.6120 | 6 | 6 |

Tableau C.5 – Résultats détaillés de l'expérience 2, $c = 8$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 45.3491 | 13 | 11 | 6 | 24.0397 | 5 | 5 |
| 2 | 40.8823 | 8 | 8 | 7 | 19.1492 | 1 | 1 |
| 3 | 34.3930 | 7 | 7 | 8 | 86.1430 | 24 | 21 |
| 4 | 24.9463 | 3 | 3 | 9 | 192.3848 | 28 | 23 |
| 5 | 143.4483 | 25 | 19 | 10 | 33.5271 | 8 | 8 |

Tableau C.6 – Résultats détaillés de l'expérience 2, $c = 10$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 23.2484 | 4 | 4 | 6 | 36.3153 | 10 | 10 |
| 2 | 41.0533 | 12 | 9 | 7 | 18.3193 | 3 | 2 |
| 3 | 22.7916 | 5 | 5 | 8 | 111.5511 | 23 | 21 |
| 4 | 71.2479 | 13 | 11 | 9 | 31.6825 | 10 | 7 |
| 5 | 186.3901 | 28 | 27 | 10 | 397.5264 | 45 | 39 |

Tableau C.7 – Résultats détaillés de l'expérience 2, $c = 12$

| # | E_{100} | ν_{100} | ν_m | # | E_{100} | ν_{100} | ν_m |
|---|-----------|-------------|---------|----|-----------|-------------|---------|
| 1 | 15.5174 | 3 | 3 | 6 | 59.2171 | 12 | 11 |
| 2 | 119.5098 | 18 | 16 | 7 | 118.0065 | 17 | 13 |
| 3 | 263.4773 | 24 | 24 | 8 | 254.7055 | 28 | 26 |
| 4 | 61.6711 | 13 | 12 | 9 | 18.0496 | 5 | 4 |
| 5 | 78.7203 | 13 | 9 | 10 | 16.4812 | 5 | 3 |